August 1980
$2.95

kilobaud

MICROCOMPUTING

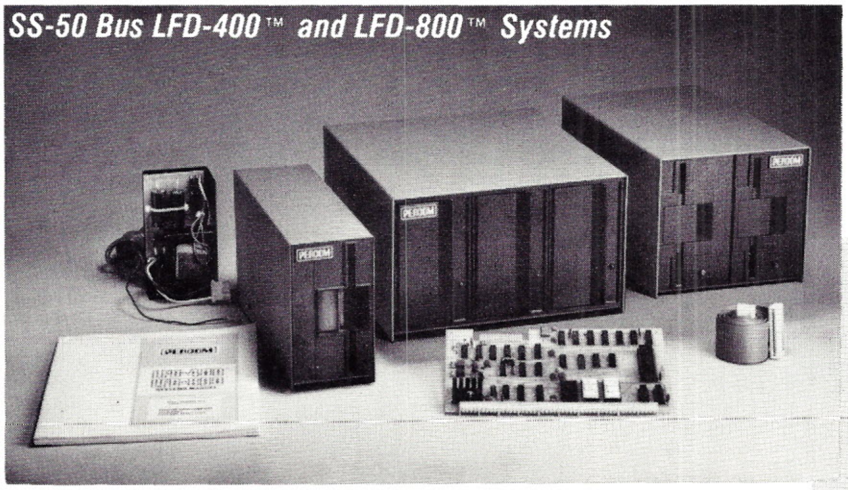
T.M.

DON LANCASTER TELLS HOW TO WIN THE MICRO GAME. P. 36

80-COLUMN DRAGON SLAIN: WIDE-SCREEN WIZARDRY FOR APPLE II. P. 100

THE ESSENTIALS OF EFFICIENT PROGRAMMING. P. 198

A Few Extraordinary Products for Your 6800/6809 Computer

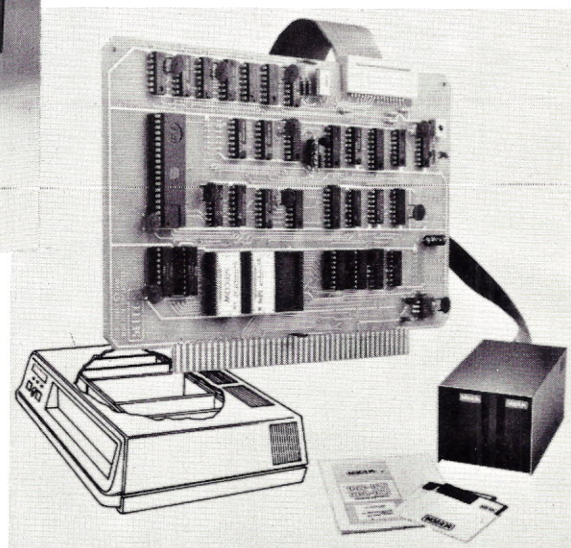


From Percom . . .

Low Cost Mini-Disk Storage in the Size You Want ✓14

Percom mini-disk systems start as low as $599.95, ready to plug in and run. You can't get better quality or a broader selection of disk software from any other microcomputer disk system manufacturer — at any price!

Features: 1-, 2- and 3-drive systems in 40- and 77-track versions store 102K- to 591K-bytes of random access data on-line • controllers include explicit clock/data separation circuit, motor inactivity time-out circuit, buffered control lines and other mature design concepts • ROM DOS included with SS-50 bus version — optional DOSs for EXORciser* bus • extra PROM sockets on-board • EXORciser* bus version has 1K-byte RAM • supported by extended disk operating systems; assemblers and other program development/debugging aids; BASIC, FORTRAN, Pascal and SPL/M languages; and, business application programs.



EXORciser Bus LFD-400EX™ -800EX™ Systems*

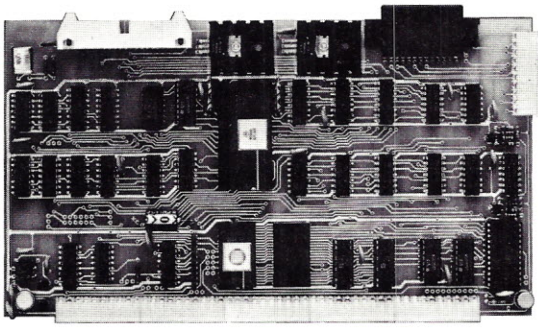
Versatile Mother Board, Full-Feature Prototyping Boards ✓15

Printed wiring is easily soldered tin-lead plating. Substrates are glass-epoxy. Prototyping cards provide for power regulators and distributed capacitor bypassing, accommodate 14-, 16-, 24- and 40-pin DIP sockets. Prototyping boards include bus connectors, other connectors and sockets are optional.

MOTHERBOARD — accommodates five SS-50 bus cards, and may itself be plugged into an SS-50 bus. Features wide-trace conductors. Price: $21.95

SS-50 BUS CARD — accommodates 34- and 50-pin ribbon connectors on top edge, 10-pin Molex connector on side edge. Price: $24.95.

SS-30 BUS CARD — 1¼-inch higher than SWTP I/O card, accommodates 34-pin ribbon connector and 12-pin Molex connector on top edge. Price: $14.95.



The SBC/9™. A "10" By Any Measure. ✓13

The Percom SBC/9™ is an SS-50 bus compatible, standalone Single-Board Computer. Configured for the 6809 microprocessor, the SBC/9™ also accommodates a 6802 without any modification. You can have state-of-the-art capability of the '09. Or put to work the enormous selection of 6800-coded programs that run on the '02.

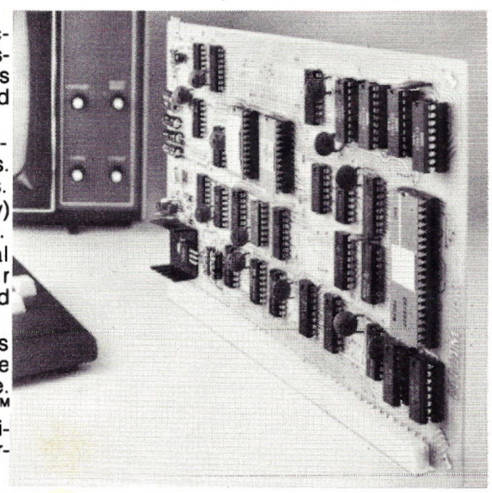
The SBC/9™ includes PSYMON™, an easily extended 1-Kbyte ROM OS. Other features include:

- Total compatibility with the SS-50 bus. Requires no changes to the motherboard, memory or I/O.
- Serial port includes bit-rate generator. RS-232-C compatible with optional subminiature 'D' connector installed. 10-pin Molex connector provided.
- Eight-bit, non-latched, bidirectional parallel port is multi-address extension of system bus. Spans a 30-address field; accommodates an exceptional variety of peripheral devices. Connector is optional.
- Includes 1-Kbyte of static RAM.
- Costs only $199.95 with PSYMON™ and comprehensive users manual that includes source listing of PSYMON™.

The Electric Window™: Instant, Real-Time Video Display Control ✓16

Memory residency and outstanding software control of display format and characters make this SS-50 bus VDC card an exceptional value at only $249.95. Other features:

- Generates 128 characters including all ASCII displayable characters plus selected Greek letters and other special symbols.
- Well-formed, easy-to-read 7x12-dot characters. True baseline descenders.
- Character-store (display) memory included on card.
- Provision for optional character generator EPROM for user defined symbols.
- Comprehensive users manual includes source listing of Driver software. Driver — called WINDEX™ — is also available on minidiskette through the Percom Users Group.



™ trademark of Percom Data Company, Inc.
* trademark of the Motorola Corporation.
Prices and specifications subject to change without notice.

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

Products are available at Percom dealers nationwide. Call toll-free, 1-800-527-1592, for the address of your nearest dealer, or to order direct.

SUPER BRAIN QD™

Once in a great while someone comes along with a simple improvement for an already great product. Take our SuperBrain, for example. Really a simple concept. A high-powered, low cost microcomputer packaged in an attractive desk top cabinet. So how do you improve on that?

WE DID IT...

It wasn't enough that our SuperBrain had such standard features as twin double density 5¼" drives with over 300,000 bytes of disk storage. A full 32K of dynamic RAM - expandable to 64K in seconds. A CP/M* Disk Operating System which assures compatibility to literally hundreds of application packages presently available. A crisp, 12" non-glare screen with a full 24 line by 80 column display. A full ASCII keyboard with a separate keypad and individual cursor control keys. Twin RS232C serial ports for fast and easy connection to a modem and/or a printer. And, dual Z80 processors which operate at 4 MHZ to insure lightning-fast program execution. No, it wasn't enough. So we made it better.

ANNOUNCING SUPERBRAIN QD...

Our new QD model has all of the features of our phenomenally popular SuperBrain with the addition of double-sided disk drives and an extra 32K of dynamic RAM. So, for only a modest increase in price, you can order your next SuperBrain with more than twice the disk and memory storage capability. But, best of all, the new QD model has the same tough, rugged construction and exceptional quality that made our SuperBrain such a success.

HOW DID WE DO IT?

The secret of SuperBrain QD's incredible disk storage lies within our new double-density double-sided disk drives. A total of nearly 720,000 bytes of data are formatted on two specially designed 5¼" drives. And that's more than enough to get you started with most serious small business applications. And SuperBrain QD's standard 64K of dynamic RAM will handle even the most complicated programming tasks.

Of course, if you're into megabytes instead of kilobytes, you may think neither SuperBrain is right for you. Not so! Intertec offers 20-96 megabytes of hard-disk storage which connects in seconds to either the SuperBrain or SuperBrain QD. So, your original investment is always protected. As you grow. No matter how much your needs expand.

BUT IS IT RELIABLE?

Our best salesmen are our present users. Not only have SuperBrain users been impressed with the inherent reliability of the system, they tell us that no other microcomputer system available today offers such a unique modular design concept. Just about the only tool required to easily maintain the system is a common screwdriver. And Intertec's total commitment to product service and customer support, with service outlets in most major cities, insures your original investment will be a valuable one for many years to come.

THE DECISION IS YOURS.

Whether your next SuperBrain is a regular model or our QD version, you will have the satisfaction of knowing you purchased what is becoming one of the world's most popular microcomputer systems. And regardless of which model you choose, you'll probably never outgrow it because you can keep expanding it.

So, call or write us today for more information. Intertec systems are distributed worldwide and may be available in your area now.

INTERTEC DATA SYSTEMS®

2300 Broad River Rd., Columbia, SC 29210
(803) 798-9100 TWX: 810-666-2115

✓196

*Registered trademark of Digital Research, Inc.

QUALITY
THAT'S WHAT SEPARATES THIS PRINTER FROM THE TOYS

MICROTEK MT-80
SOLID VALUE FOR YOUR DOLLAR

The market is flooded with low-cost printers that look and last more like toys.

The Microtek MT-80, our versatile alphanumeric line printer, has a high quality print mechanism that gives you solid value for your dollar. It has been designed with a superior brain resulting in more advanced features and more dependable performance. Our printer is so reliable that we offer you an incredible 365 days warranty.

We stand behind every printer we make because we build quality into each one. So stop tinkering with toys and get serious. Demand 100% value by specifying the MT-80.

OUR UNIT PRICE
\$795 Parallel
\$895 Serial (RS-232C)

LOADED WITH INNOVATIONS
- 40, 80 or 120 columns (software selectable)
- Non-thermal paper, pin feed
- 125 CPS, 70 lines per minute
- 9 × 7 dot matrix
- Vertical format unit
- 96-character ASCII (upper and lower case)
- Adjustable forms width to 9½″
- Parallel and serial (RS-232C) interfaces available

MICROTEK inc.

347

For more information contact:
MICROTEK, Inc.,
9514 Chesapeake Drive,
San Diego, CA 92123
Tel.(714)278-0633
TWX 910-335-1269

kilobaud

MICROCOMPUTING™

contents: august '80

ARTICLES

DEPARTMENTS

Cover illustration by Dion Owens.

micro info

$ This symbol next to a title in the table of contents indicates that the article is a business-application article.

Manuscripts

Contributions in the form of manuscripts with drawings and/or photographs are welcome and will be considered for possible publication. We can assume no responsibility for loss or damage to any material. Please enclose a self-addressed, stamped envelope with each submission. Payment for the use of any unsolicited material will be made upon acceptance. All contributions should be directed to the *Microcomputing* editorial offices. "How to Write for Microcomputing" guidelines are available upon request.

Editorial Offices:

Pine Street
Peterborough NH 03458
Phone: 603-924-3873

Advertising Offices:

Elm Street
Peterborough NH 03458
Phone: 603-924-7138

Circulation Offices:

Elm Street
Peterborough NH 03458
Phone: 603-924-7296

To subscribe, renew or change an address:

Write to *Microcomputing*, Subscription Department, PO Box 997, Farmingdale NY 11737. For renewals and changes of address, include the address label from your most recent issue of *Microcomputing*. For gift subscriptions, include your name and address as well as those of gift recipients. Postmaster: Send form #3579 to *Microcomputing*, Subscription Services, PO Box 997, Farmingdale NY 11737.

Subscription problem or question:

Write to *Microcomputing*, Subscription Department, PO Box 997, Farmingdale NY 11737. Please include an address label.

Kilobaud *Microcomputing* (ISSN 0192-4575) is published monthly by 1001001, Inc., 80 Pine St., Peterborough NH 03458. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada: $27 for one year only, U.S. funds. Foreign subscriptions (surface mail)—$35 for one year only, U.S. funds. Foreign air mail subscriptions—$62 for one year only, U.S. funds. In Europe, contact: Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. South African Distributor: KB Microcomputing, PO Box 782815, Sandton, South Africa 2146. Australian Distributor: Electronic Concepts, Attention: Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Phone: 603-924-3873.

Wayne Green

PUBLISHER'S REMARKS

Keyboarding Economy

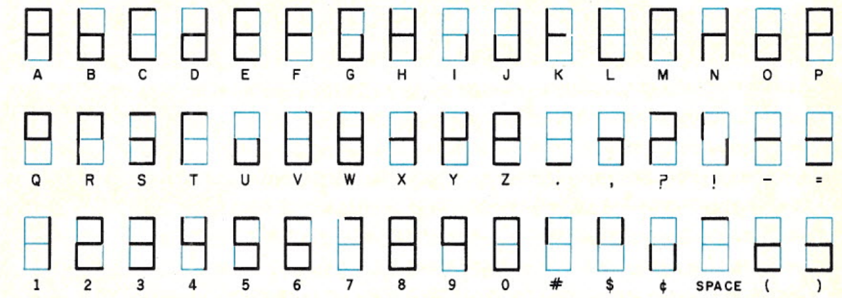


Fig. 4. This is the entire alphabet, numbers and some punctuation in the Green Readable Character set. This uses just the seven-segment readouts.

The postal service reportedly poured millions of dollars down the tubes trying to invent the uninventable: a machine for deciphering handwriting. Unquestionably there is a need for this.

Most of the data input to computers consists mainly of names, addresses and orders for goods. There seems to be no practical approach to induce people to buy a typewriter with a special ball that can be read by optical character-reading (OCR) systems. The postal service might be able to implement some sort of OCR addressing if they set up a special class of mail with lower rates for letters addressed with a standard OCR.

The answer to speeding up (cutting the cost) of data input for computers lies in some sort of automatic reading of names, addresses and other simple data such as quantity and part number for orders. We already have equipment that reads pencil or pen marks on cards, so that gives us something to work with. Since there are too many ways of writing or printing for a computer to cope with, we are going to have to come up with some simplified coding which our computers can read, but which can be marked with a pencil or pen.

There are about 40 different characters that need to be deciphered (alphabet, numerals, punctuation marks). A system that required you to choose each individual character from a group of 40 boxes, each box representing one character, would be too cumbersome. Since a name and address would have to allow for up to 80 individual characters, a coded card would have to contain 3200 boxes just to cope with the name and address. Something simpler is necessary, and I think I have a solution.

Most of my flashes of intuition come while I am taking a shower. (I've read that this has something to do with the negative ions caused by the stream of water hitting the body. There may be something to this. I don't take long showers, either—though I suppose I should. I've worked my morning shower process into an efficient system: brushing my teeth (yes, in the shower, while the beard is softening), shaving (yes, also in the shower, while the beard is soft and very easy to shave), soaping and rinsing. The whole works takes under ten minutes, including toweling, applying after-shave and antiperspirant and combing what's left of my hair. If I save only ten minutes a day with this routine, it gives me 60 hours a year to do more productive things.)

This time the brainstorm had to do with utilizing the seven-segment LED or LCD character for an OCR system. With seven segments we should be able to represent 2^7 characters, or 128. That's plenty for the alphabet, numbers and punctuation.

Should we go to an ASCII-type coding or start over again? Since lookup tables for electronic circuits are so simple, the coding should be easy to read. If we start with the ten numbers, which are all handled easily with the seven-segment system, we are off and running (see Fig. 1). The alphabet is more difficult, but let's look at the letters that are easy to handle first: A, C, E, F, G, H, J, L and P (see Fig. 2).

Some letters, such as a lowercase b and d, can be handled without too much strain. Half-sized i, o and u finish off the vowels, so they are all readable (see Fig. 3). I tackled the rest of the letters and punctuation, and I came up with Fig. 4.

To see if this was worth the trouble, I tried a few words in this new representational alphabet (see Fig. 5). It's a bit of a strain, but I think that a data-input person could become accustomed to reading this in a day or less.

The end result would be one seven-segment box for each character. A translation for the marks would not take up a lot of room on a coded order card, and the finished product could then be read by machine. This would cut the cost of data entry, and thus make it possible to sell products or subscriptions cheaper. Remember that many data-input systems call for each name and address to be keyboarded a second time for verification, so perhaps you can appreciate the savings.

If those savings are not apparent, then look at it this way. At one minute per order for data entry, plus a second minute for verification of the original data, we're looking at around 14¢ in data-entry costs just for labor. Add the cost of the equipment, worker benefits, heat, light, power, repairs, training and supervision, and you at least double your cost of labor, bringing the data input cost up to 28¢ per order. Wouldn't you like to save 28¢ on every subscription to a magazine you send in . . . on every mail order you send in? An outfit that handled 1000 orders a day could save $280 per day by changing to GRC (Green Readable Character) data input. That's over $70,000 a year.

With this GRC system, a mail-order house or subscription agency would only need two data-input people. One person would take a quick look at each data input to make sure it was not screwed up; a second person would input those orders too badly confused for the system.

I doubt if this system would put a lot of people out of work. Most data-input departments I've seen are chronically short of people—not many employees enjoy sitting at a tube all day long typing in orders or reader-service requests.

This system should also be adaptable to post-office tasks. The first readers would be for

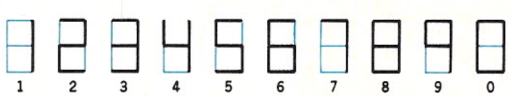


Fig. 1. Green Readable Character numbers. These are almost identical to standard seven-segment readouts.

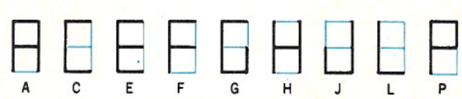


Fig. 2. Some of the letters of the alphabet adapt to being easily read in seven-segment readouts.

Fig. 3. The vowels can be made simple to read.

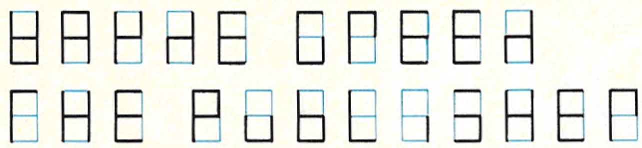


Fig. 5. With a few minutes' practice almost anyone can read this character set. It isn't great, but it is handy.

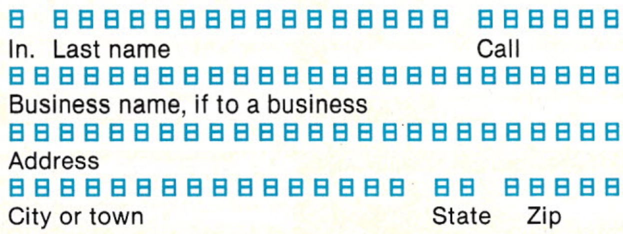


Fig. 6. This address block can accommodate a ham subscription or, by changing the call letter block to a business title, a business subscription to a magazine or on an order form. If substantial foreign business is expected, this will have to be modified to handle the zip code systems and allow a place to indicate the country.

postcard-size cards with an optically readable synchronizing line for the system to use as a reference to start reading the name, address, city, state, zip and other order information. That machinery would adapt to read addresses on envelopes and automate letter mail.

Light green outlines for the characters could be printed on the envelopes or on the labels to stick on the envelopes. Characters would be marked by pencil or pen over the green guidelines. If the postal office used reader-sorters, they could drop the postage rate to maybe 10¢ for GRC mail, and soon every business in the country would use it, complete with GRC balls for typewriters.

Using the size and spacing of the IBM Selectric typewriter, a complete name and address could fit in an area 2½ inches long by 1 inch high, and that would provide three lines of 25 characters each. A four line address system would be 1⅓ inches high. That's within the size requirements for presently used labels, so the system should not be particularly difficult to put into operation.

A character reading system such as this could be adapted to any computer, either by use of a chip with the lookup table in it or a simple lookup software program. An operator watching cards being read would see the characters translated into regular terminal characters rather than seven-segment readouts.

The character reader could either scan a row of characters, reading them sequentially, or have 875 sensing elements to read the four 25-character lines at one time. Another element or two could detect synch indicators. Order-reading systems would probably scan more like bar-code systems.

The standard I have set for the system is ten characters to the inch and a pitch of two points per line (⅓ inch). The character blocks are ten point in size and can be approximated by using the IBM #10 Orator ball and typing an E overprinted with an H. This leaves a line beneath each character line for indicating what should be on that line.

Hitachi and Tandy Coming?

There are hints that Hitachi is releasing a new system that could sell well in this country. I understand that it has eight colors and a 640 × 200 resolution. It uses a 6809 processor.

We're still waiting for the release of the Tandy color system, rumored to be called TRS-90 and use a Z-80. Although Tandy had plans for using the 6809, I suspect that the importance of being able to use the already developed program base had some effect on the decision. One of these days, we'll see where rumor ends and facts begin.

The first evaluation units of the Hitachi system should arrive in this country by early fall. I think we're in line to look at a system, so watch for more news.

Winchester Technology

This term is used a lot these days, and I'll bet that few readers have any idea of its origin. It all started, according to a recent Pertec newsletter, with a 1973 IBM 3340 drive. This new disk development was supposed to be a dual 30 megabyte drive: 30-30. The drives eventually became 35 and then 70 megabytes each, but by then the "Winchester" name had stuck. So much for the history lesson for today.

(continued on page 218)

Sherry Smythe

OUTPUT FROM ISI

Program Search

Exidy's acquisition by Recordec should increase demand for program packages for this system. ISI is particularly interested in Exidy-oriented programs for possible publication. If you have both a TRS-80 and an Exidy, you might like to try translating programs from one system to the other—for a royalty on the sales.

North Star is doing well, and ISI has North Star-based program packages well along in development . . . and needs more. If you have written any good North Star programs, you can help both North Star and yourself by submitting them for possible publication.

It will be quite a while before we have a serious need for Apple III programs, but the ISI programs currently distributed for the Apple are faring well and are a nice source of income for the authors. Now that ISI has a good foundation of TRS-80 programs, more time is being invested in the support of the Apple and other systems, so turn your Apple programming work into royalty income.

We also need a machine-language word processor—particularly for the TRS-80. This would speed up the system and make such a use for the TRS-80 far more practical. How about it, all you machine-language fans?

Translators Needed

Pressure is being put on microcomputer manufacturers to provide software support for their systems: some of the majors are beginning to feel the heat. Dealers are complaining, and even the general-interest magazines are mentioning poor software support. Money magazine quoted Texas Instruments as having plans for only 100 programs by the end of 1981. This tiny number has to put a chill on the whole industry.

It seemed appropriate for Instant Software to help by putting some TI programs on the market. We consulted our associate-editor list to see who had volunteered to make the conversions from the TRS-80 the TI-99/4. With over 1,000 editors from which to choose, we could find none available for this job!

Readers who want to tackle this should write and give me some idea of their equipment and experience. We have released hundreds of programs for the TRS-80; we want to translate as many as possible for use on the TI-99/4. We understand the problems with the system and the efforts Texas Instruments made to make translations difficult.

Several other firms have shown interest in getting substantial software support, so we may need translators for programs from TRS-80 to Heath, Atari, Apple and a few others. You do the work on your time, at your convenience. If you think you might go for this, drop me a line.

BOOK REVIEWS

Electronic Design with Off-the-Shelf Integrated Circuits

Z. H. Meiksin and Philip C. Thackray
Parker, West Nyack, NY, 1980
Hardbound, 383 pp., $19.95

This cookbook contains the design principles you'll need to exploit low-cost mass-produced integrated circuits and passive support components. An excellent one-stop reference for the computerist who designs his own hardware, the book is for both those who are only adding to their systems and those who are full-blown construction freaks.

The text focuses on practical approaches. It minimizes math and emphasizes a cut-to-fit method that must, in the end, be applied to the most thoroughly planned designs.

For example, the book supplies simplified equations for designing active filters with op amps. It covers first through fourth order, constant ripple, maximally flat and high-Q filters. Even with the simplified tables and equations provided, the authors say, values that result are "of much greater precision than can be obtained with commercial fixed value components." The text then outlines filter tuning procedures.

While most readers will use the sections on digital ICs primarily for reference, topics such as rise time and decay time are presented in an unusually lucid manner. The chapters on analog to digital conversion, linear applications and nonlinear applications are also thought-provoking. These chapters cover photo sensor amplifiers, four-quadrant multipliers, peak detectors and sample and hold circuits.

The book, unlike any other text I've seen, covers passive components with the same depth it covers active components. These include resistors, capacitors, inductors and transformers.

This information will help you decide what type or composition of capacitor to use or what type of resistor you'll need. The characteristics of each type of part are explained with enough detail to let you make intelligent decisions.

The authors also discuss techniques for anticipating and eliminating noise problems. Computer buffs will be particularly interested in the chapter on grounding and shielding. A section on eliminating spurious signals in digital circuits includes ways to handle power supply noise, stray field effects, capacitive coupling and signal reflection. The chapter also suggests ways to overcome problems with analog and hybrid digital-analog systems.

The text ends with examples of system design and summarizes considerations that affect the overall performance of a system. The book provides guidelines for choosing the IC functional blocks that best satisfy an application's requirements with the greatest cost-efficiency.

"Pitfalls to Avoid," a feature of each section, has already earned the price of the book for me. It covers common errors in the design approach to applying the particular component. For example, the authors point out a common error in the design of light detector amplifiers and suggest an approach that eliminates the problem and reduces the circuit's part count.

While much of the book's information is

duplicated elsewhere in my library, nowhere is it so conveniently brought together. I think that if you have any experience with building or designing hardware, you'll find this a useful text that is neither too complicated nor too simple.

Gregg W. Squires
Sparkill, NY

Microcomputer Interfacing

Bruce A. Artwick
Prentice-Hall, Englewood Cliffs, NJ
Hardbound, 323 pp., $18.95

The title of this book led me to believe that it would discuss how to attach peripherals or signal conditioners to a functioning computer system. But it instead tackled such topics as selection of microprocessor chips, design rules for microprocessor controllers and construction techniques. Artwick is clearly talking about microprocessor interfacing.

If you're a hobbyist ready to become an entrepreneur, your first decision will be which microprocessor to choose. Four, eight or 16 bits? High speed or low power? One whose instruction set is directed toward data processing or industrial control? Artwick compares about a dozen different microprocessors. He is uneven, but does touch on the weaknesses and strengths of each. He also includes bit-sliced chips and single board controllers. Novice designers will find this helpful.

Because of its clear explanation of both static and dynamic RAM, the chapter on data storage is the best in the book. Imagine someone boiling down volumes of manufacturers' literature and making the residue readable. Starting with the old 2102 memory chip and using many simplified block and timing diagrams, he works his way through to the recent dynamic 4 and 16K RAMs. The text answers such questions as "Why do the new dynamic 16K RAMs have fewer address pins than 1K static RAMs?" and "What is the difference between an erasable PROM and an alterable ROM?" It even looks briefly at charge-coupled devices and magnetic bubble memory.

About one quarter of the chapter deals with magnetic recording methods for long-term storage on tape or floppy disks. Hard disk technology is ignored.

The chapter on interfacing components also deserves special mention for its completeness. Any possible way to get information into or out of a computer system is listed. Artwick covers the basics of TTL drivers and receivers and explains the importance of fan-out, what to do about unused gate inputs and how to mix MOS and TTL family chips in the same design. This is followed by the LSI interface chips, such as programmable peripheral interface chips and CRT controllers.

A good section explains why a synchronous communication interface adapter is more complicated than an asynchronous communication interface adapter. The pages on analog and digital conversion, transducers and optical displays contain many practical suggestions and criteria for selection.

Each topic, however, is limited in scope. A section on floppy disk controller chips mentions two that are designed to mate with specific microprocessors but neglects to mention the Western Digital 1790 series, which is manufactured for use with any microprocessor.

The rest of the text is more general. The chapter on input and output methods spends less than a page each on asynchronous serial communications, multiplexing and memory-mapped I/O and doesn't include examples. Artwick's coverage of polling and interrupt-driven I/O is a bit more comprehensive, because he offers suggestions on how to assign interrupt priorities and handle interrupt processing.

Other chapters cover circuit board layout, construction techniques and interfacing to standard microcomputer buses.

Artwick does not include a bibliography. This is unfortunate, since I found many places where I needed more information. The figures are excellent and the majority of the text is very clear. The coverage of future trends in the industry will keep the book current long enough to avoid having to put out the second edition in loose-leaf form.

The overall usefulness of this book would depend on who you are. If you want to design something, *Microcomputer Interfacing* is just a starting point. It is not a circuit design reference book because not many specific circuits are included.

The book is good for hints, tips and rules. Here are two examples: "One-shots should never be used to drive Set, Clear or Clock inputs of logic devices," and "Convert analog signals into digital signals as soon as possible."

I doubt that an inexperienced designer would be able to complete an assignment with just this book, but it should shorten development time. The person who wants to know how microprocessor-based equipment works will be helped the most.

Mike Aronson
Oregon City, OR

Introduction to Computer Programming

Walter S. Brainerd
Charles H. Goldberg
Johnathan L. Gross
Harper and Row, New York, 1979
Hardcover, 534 pp., $16.95

The authors promise a textbook that will give the beginner a solid foundation of programming knowledge. And though the title is a little misleading—the novice will need to know something about programming if he does not have an instructor—the book is one of the best treatments of the art and science of programming I have used.

The authors take a unique approach. They have developed a Beginner's Programming Language (BPL) that teaches the logic of programming without requiring the reader to master the conventions of a specialized language. The beginner can read and understand a BPL program without any knowledge of the BPL rules; the language itself carries the structure of its own logic.

The beauty of this approach is the ease with which a student of BPL can transfer his learning to BASIC, FORTRAN, PL/1 and COBOL programming. Once the reader adjusts to BPL, the 200 sample programs found in the text are completely understandable.

The book moves from the simple to the complex, using examples and applications every step of the way. For example, a continuing analysis of credit card sorting presents an excellent tutorial of various sorting routines. The authors go through finding the smallest element in a list, sorting by replacement, merging, the bubble sort and, finally, file sorting.

As they present this material, the authors lead the reader through formatting, looping and other standard tools of programming. The beginner learns in a building-block fashion, each bit of knowledge leading to the next.

The beginning programmer should follow the text sequentially. However, a reader with a solid foundation in programming will want to study selected chapters. The 500-plus exercises throughout the book provide a constant source of stimulation and challenge.

James P. Morgan
Scott AFB, IL

NEW RELEASES

6502 Software Design—Leo J. Scanlon. Howard W. Sams & Co., Inc., Indianapolis, IN. $10.50.

Programming with ADA: An Introduction by Means of Graduated Examples—Peter Wegner. Brown University, Providence, RI. $13.95.

Programming & Interfacing the 6502—Marvin L. DeJong. Howard W. Sams & Co., Inc., Indianapolis, IN. $13.95.

Microprocessors/Microcomputers/System Design—Texas Instruments, Inc. McGraw-Hill, New York. $24.50.

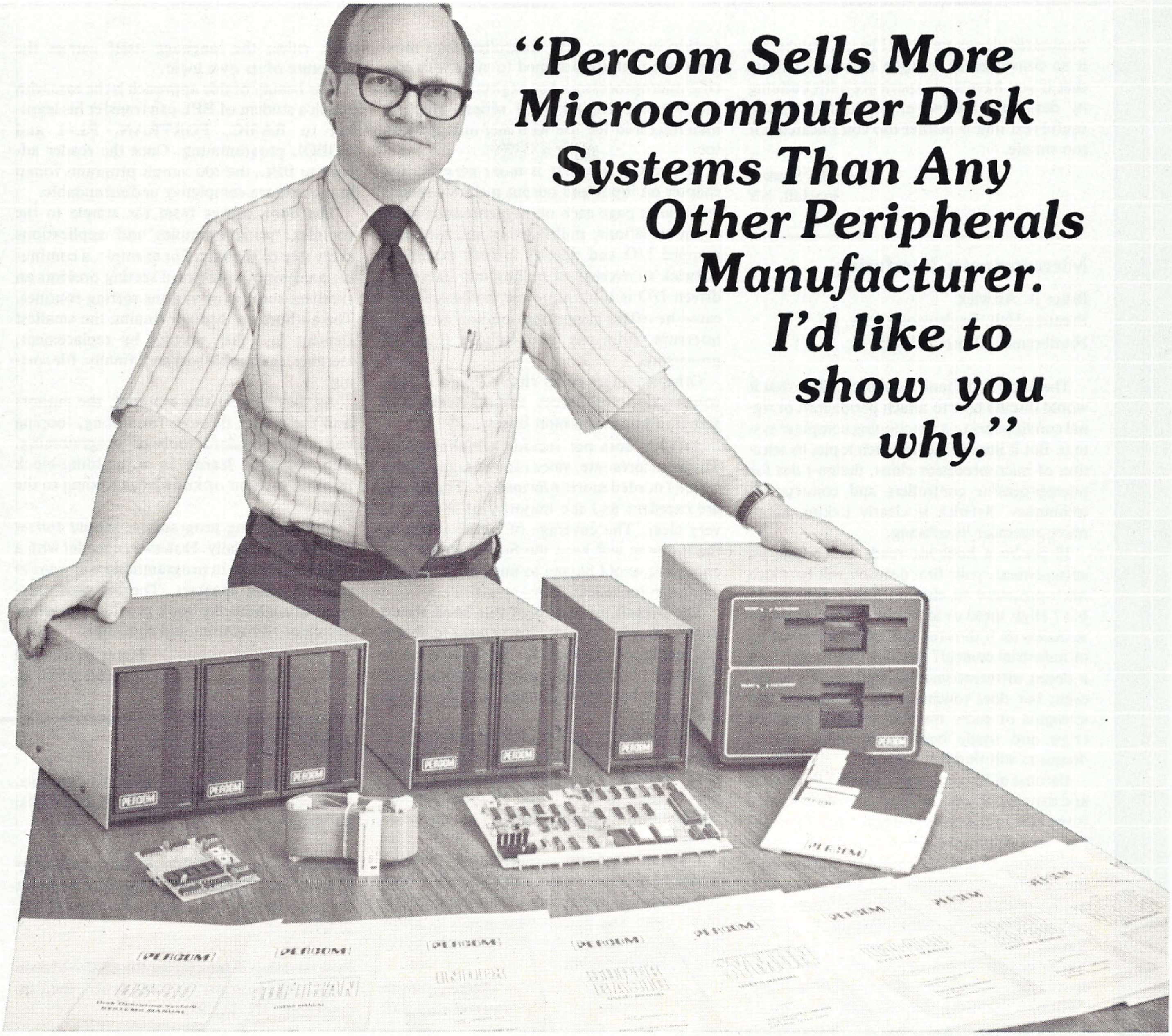
The Howard W. Sams Crash Course in Microcomputers—Louis E. Frenzel, Jr. Howard W. Sams & Co., Inc., Indianapolis, IN. $17.50.

Troubleshooting Microprocessors and Digital Logic—Robert L. Goodman. TAB Books, Blue Ridge Summit, PA. $12.95, hardbound ($7.95, paperback).

Computer Consciousness: Surviving the Automated 80s—H. Dominic Covvey and Neil H. McAlister. Addison-Wesley, Reading, MA. $5.95.

A Bit of BASIC—Thomas A. Dwyer and Margot Critchfield. Addison-Wesley, Reading, MA. $5.95.

Experimenter's Guide to Solid State Electronics Projects—Alfred W. Barber. Parker Publishing, W. Nyack, NY. $14.95.



"Percom has been manufacturing mini-disk storage systems for microcomputers since 1977 when we introduced the 35-track, single-drive LFD-400™. Now we produce 1-, 2- and 3-drive systems in 40- and 77-track versions, a multi-density MEGABASE™ system and a host of accessories and software.

"Volume not only means experience in critical production and testing operations, it also means we can offer superior design features, extra testing and qualified backup support at very competitive prices.

"I know of no other microcomputer disk system manufacturer who even begins to offer the broad spectrum of disk equipment and programs available from Percom."

"So before you buy a mini-disk system for your 6800, 6809 or TRS-80* computer, take a good look at what the people at Percom have to offer."

Harold Mauch
President, Percom Data Company

Percom disk systems start at only $399.00. Disk systems and other quality Percom products are available at computer dealers nationwide. **Call toll-free, 1-800-527-1592, for the locations of dealers in your area, or to order direct.**

↙17

PERCOM

PERCOM DATA COMPANY, INC.
211 N. KIRBY GARLAND, TEXAS 75042
(214) 272-3421

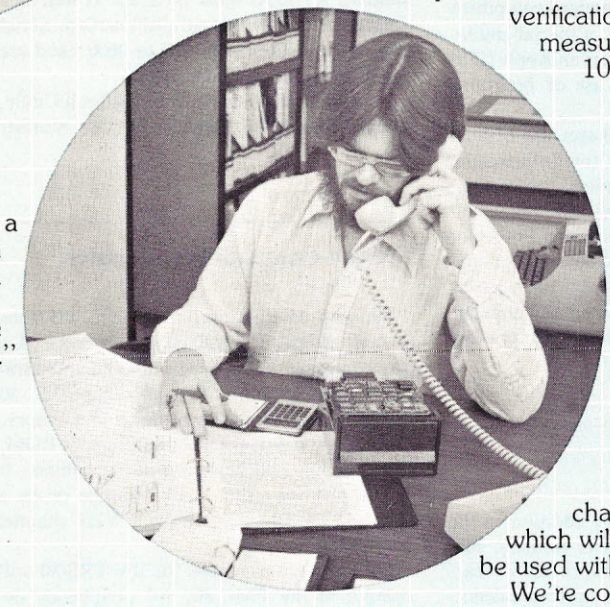
PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

TM trademark of Percom Data Company, Inc.

*trademark of Tandy Radio Shack Corporation which has no relationship to Percom Data Company.

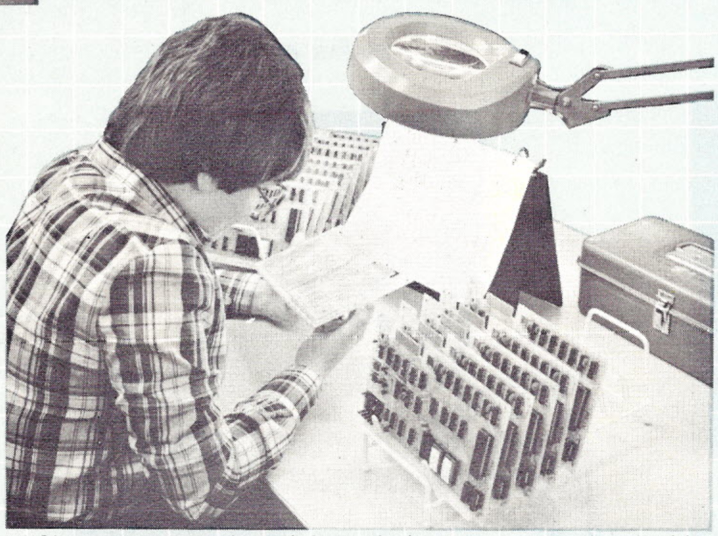
"From an efficient 1K-byte control system DOS to high level languages such as FORTRAN and Pascal, no other microcomputer disk systems manufacturer provides the range and quality of development and application programs available from Percom."

"Connie is running a 'cats eye' test on a mini-disk drive to check radial track alignment. Drive motor-speed timing and sensor alignment tests have already been performed. Disk formatting and format verification tests are next. These measurements are part of the 100% testing every single unit receives."

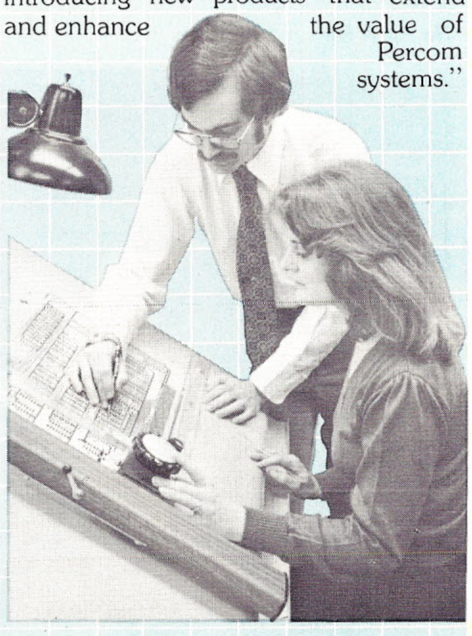


"Whether you call about a shipping date or ask a tough technical question, you get a competent courteous answer. Outstanding customer service is a hallmark of Percom."

"Richard's making final changes to a disk controller which will allow Percom drives to be used with yet another computer. We're constantly developing and introducing new products that extend and enhance the value of Percom systems."



"Slipping a circuit board through the eye of a needle would be easier than slipping a cold solder joint past Beverly. These are four-drive LFD-400/800 disk system controllers she's inspecting."



Edited by Dennis Brisson

NEW PRODUCTS

Apple III

Apple Computer, Inc., 10260 Bandley Drive, Cupertino, CA 95014, has introduced its Apple III computer. Designed for use by professional/managerial personnel, the Apple III features a new Apple-designed central processor, up to 128K bytes of main memory, a built-in disk controller for handling up to four floppy disk drives, a new keyboard design with a 13-key numeric keyboard and an 80-character × 24-line upper/lowercase display. A number of items that were optional on the Apple II have been incorporated as standard equipment in the Apple III. In addition, a special emulation capability lets users convert an Apple III to an Apple II to permit the use of programs developed for the Apple II.

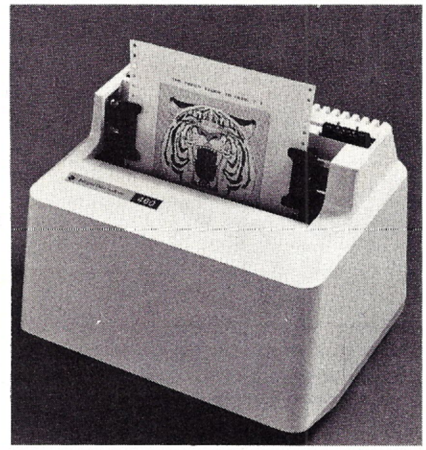
Two new application packages are offered for use on the Apple III: the Information Analyst software, for planning, forecasting, modeling, pricing and costing, scheduling and budgeting, and the Apple III Word Processor software, for preparing memos, letters and general typing, long documents, form letters and legal documents. Apple III prices range from $4340 to $7800. Reader Service number 481.

Dot-Matrix Printer

The Model 460, the newest addition to the Paper Tiger line of dot-matrix printers, features bidirectional print speeds of 160 characters per second and offers correspondence-quality, high-speed printing and high-resolution graphics. It offers a variety of programmable print control functions including proportional character spacing and automatic text justification.

The Model 460 can print in 80-, 96- and 132-column formats. Standard paper-handling features include adjustable pin-feed tractor drives that use a stepper motor to ensure fast, accurate movement of fanfold or roll paper and single- or multi-part forms ranging from 1.75 to 10 inches wide. A standard 2K byte buffer allows acceptance of the entire contents of a full 1920-character CRT screen. The unit has a standard RS-232C serial interface as well as a Centronics-compatible parallel interface. Serial transmission rates from 110 to 9600 baud are switch-selectable. Price is $1295.

Integral Data Systems, Inc., 14 Tech Circle, Natick, MA 01760. Reader Service number 477.



The latest Paper Tiger series printer from IDS.

TRS-80-Compatible Computer

Personal Micro Computers, Inc., 475 Ellis St., Mt. View, CA 94043, has introduced a new computer that is hardware- and software-compatible with the TRS-80. The PMC-80 features a cassette tape recorder, 16K memory, Level II Microsoft BASIC interpreter in ROM, power supply and keyboard in one cabinet. It will display on either a TV monitor or on a standard TV set using a built-in VHF channel 3 modulator.

All software available for the TRS-80 will operate in the PMC-80. All peripherals designed for the Radio Shack parallel port will interface to the PMC-80 50-pin bus through a 40-pin interface adapter available from PMC. Disk-based programs can be run on the PMC-80 using the Radio Shack Expansion Interface, or equivalent, which makes peripherals designed for the TRS-80 (such as Winchester disks, speech recognition, printers or RS-232 adapters) compatible with the PMC-80. Price is about $200 less than a comparably equipped TRS-80. Reader Service number 476.

Hardware Debug Aid

New Technologies Co., PO Box 32, Streamwood, IL 60103, now offers an inexpensive al-

The Apple III.

The PMC-80.

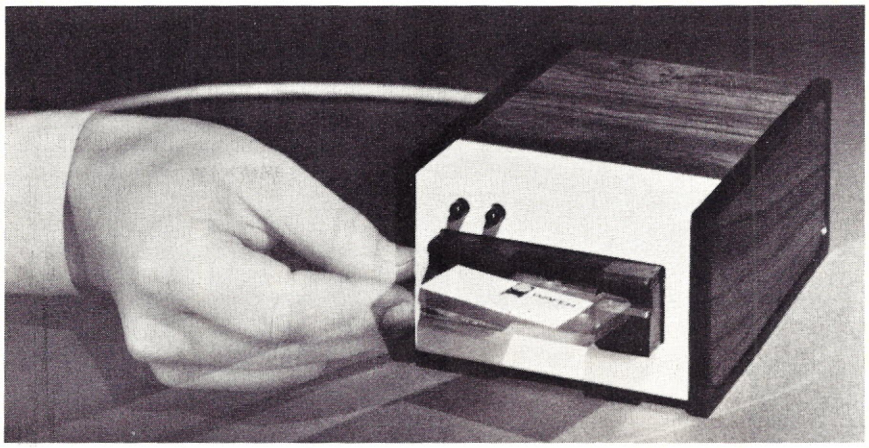
ternative to logic analyzers and logic probes in troubleshooting micros. The Hardware Debug Aid (HDA) is an S-100 board that provides sync pulses for oscilloscope use in troubleshooting specific instructions. Or, you can disconnect the address bus to sync any combination of up to 17 signals by using jumpers. The sync pulse is also used to latch and display the status of up to eight TTL-level signals. Price is $99.95 ($84.95, kit). Reader Service number 473.

RS-232 Storage Device

The Micro-Sponge is a mass storage device that is jumper selectable for 300, 1200, 9600 and 76.8K baud and stores a maximum of 80K bytes on a 75-foot Exatron Stringy Floppy wafer. The unit plugs into any computer system that has an RS-232 port. Wafers come in five-foot increments of tape length from a minimum of five feet to a maximum of 75 feet, and each five-foot length of tape stores 5.3K of RS-232 formatted data.

The Micro-Sponge features four basic commands: Read, Write, Go to Beginning of Tape and Space File Forward. The Sponge buffers up to 1000 bytes of data in internal RAM before writing out to the wafer and requires 4.5 seconds to transfer 5.3K bytes at 9600 baud—24 seconds maximum to find the beginning of tape in an average length wafer. Price is $349.50.

Exatron, 181 Commercial St., Sunnyvale, CA 94086. Reader Service number 471.



Exatron's Micro-Sponge.

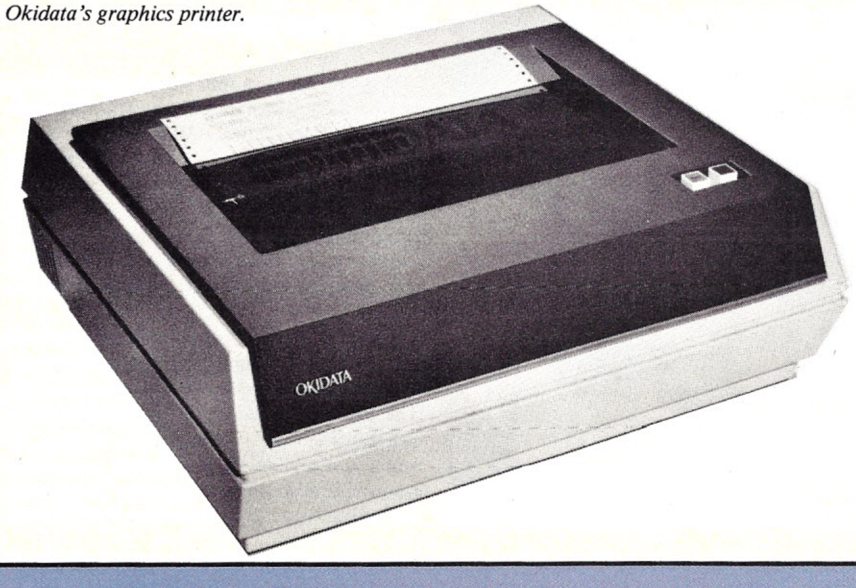
Variable Speed/Dot Density Printer

The Slimline SLG is a new graphics printer that provides a choice of two-speed/two-dot-density printing for alphanumerics. It will print routine reports at 400 lpm with a low-density pattern (7×5 and 7×6 matrices) and then switch to a high-density pattern (7×9 and 7×12 matrices) to print correspondence at 120 lpm.

In graphics mode, it provides a dot density of 100×100 dots per inch at a plotting speed of 12 inches per minute. It will reproduce anything that can be displayed on a CRT screen, including graphs, maps, bar charts and labels, as well as such foreign language characters as Arabic, Chinese and Farsi. It is available with Printronix-, Centronics- and Dataproducts-compatible parallel interfaces and with a microprocessor-controlled RS-232 serial interface.

Okidata Corporation, 111 Gaither Drive, Mount Laurel, NJ 08054. Reader Service number 475.

Okidata's graphics printer.



H89 Disk Accessory

The H77 is a new floppy disk accessory for the H89 all-in-one computer. The H89 can accommodate up to three floppy disk drives with the H77. You can run operating system and program disks at the same time for fast and efficient access to programs and data.

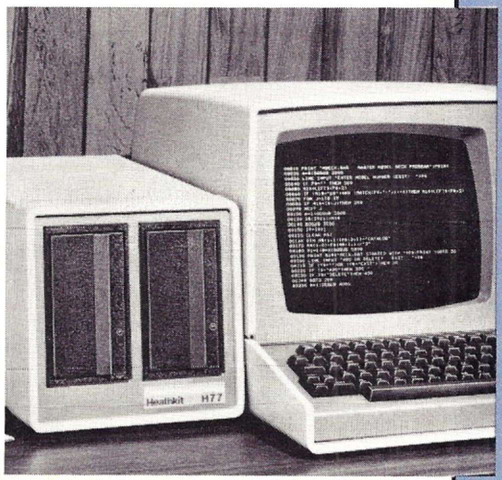
Based on the H17 Floppy Disk System, the H77 uses standard 5.25-inch, hard-sectored 40-track diskettes, each of which is capable of storing 100K bytes of data. It uses the Siemens 82 disk drive system, which provides reliable high-speed access to data. Random sector access time is less than 250 milliseconds. The H77 ($595) includes one disk drive. A diskette storage accessory, which fits into the space reserved for the second drive, is also included. The H17-1 ($325) is available to provide two-drive capability for the H77, giving the H89 a total of three drives including the drive built into the computer itself.

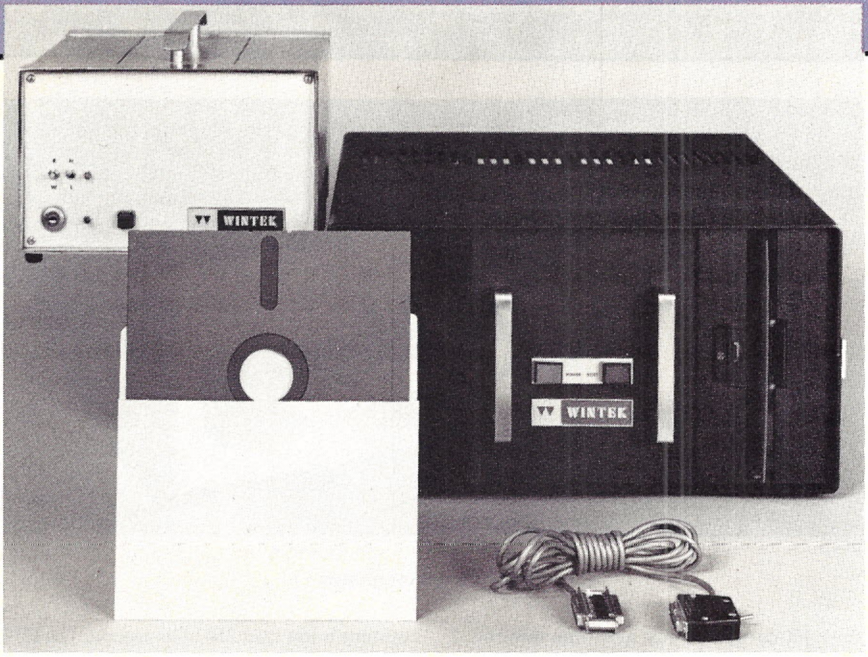
Heath Company, Benton Harbor, MI 49022. Reader Service number 483.

Single Board Computer

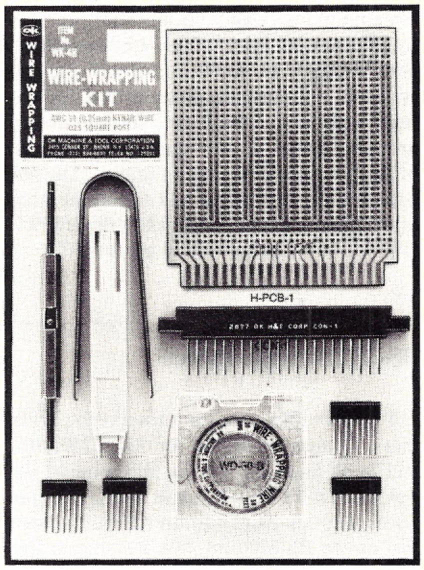
The ZCB single board computer, designed to function as the center of a unique approach to system design, is aimed at system integrators as well as the industrial process control and scientific markets. It generates all standard S-100 bus signals, including emulation of an 8080 CPU, and contains a Z-80-A operating at 4 MHz, 1K of high-speed static RAM memory, three sockets for up to 12K of PROM, one serial port and three 8-bit programmable parallel ports. Circuitry is provided to support static or dynamic memories. Use of 2708, 2716 or 2732 PROMs is jumper selectable, and the addressing of the PROM and RAM is completely variable. Use of wait states on bus cycle and/or in-

H77 Floppy Disk System.





Wintek's Down Loader system.



The WK-4B wire-wrapping kit.

struction fetch cycle is also jumper selected.

The serial port makes use of the Intel 8251 USART, which enables software to control the format of the transmitted data and to vary the mode of transmission. A DIP switch specifies the basic rate, between 110 and 9600 baud. The parallel ports use the Intel 8255, which allows the same lines to be used for input and output, under program control, and allows flexibility in assigning lines to I/O addresses, also under program control. Price is $395.

Vector Graphic, Inc., 31364 Via Colinas, Westlake Village, CA 91361. Reader Service number 480.

Microcomputer Down Loader

Wintek's new Down Loader allows you to automatically down-load programs developed on the Sprint 68 microcomputer to a target computer for final debugging in their true operating environment. The system consists of a switched RS-232 cable assembly and associated software on diskette. Price is $149.

Wintek Corp., 1801 South St., Lafayette, IN 47904. Reader Service number 470.

Wire-Wrapping Tools

OK Machine and Tool Corp., 3455 Conner St., Bronx, NY 10475, announces several new tools and parts for prototype and hobby applications.

The WK-4B wire-wrapping kit includes a universal PC board, an edge connector with wire-wrapping terminals, two industrial-quality 14-pin wire-wrapping DIP sockets, two 16-pin sockets, a DIP inserter/extractor, a wire dispenser with 50 ft. of wire and a cutting and stripping mechanism to prepare the wire for wire-wrapping or soldering and a new combination tool that wraps and unwraps 30 AWG wire on .025 square pins, plus strips 30 AWG wire using a convenient built-in stripper. Price is $25.99. Reader Service number 485.

Word Processor/Computer

Superstar is a word processor/small-business computer that consists of ITI's Superbrain by Intertec, the NEC Spinwriter and MicroPro's WordStar word-processing software. This combination features word wrapping, dynamic pagination, two double-density 5¼-inch floppy disk drives, 64K bytes of user-programmable RAM and printing at 55 cps. Price is $7500. Software necessary to handle a company payroll of up to 75 people, general ledgers, accounts receivable and inventory is available for $2500.

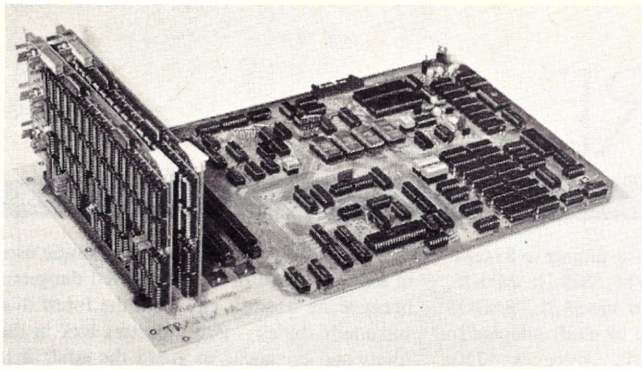
Information Technology, Inc., 56 Kearney Rd., Needham, MA 02194. Reader Service number 469.

The Superstar combination.

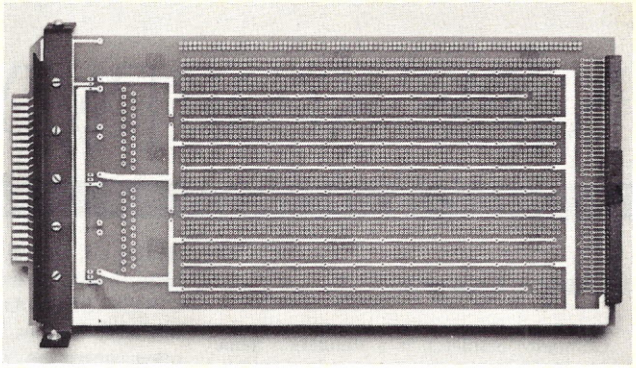
Typewriter Interface

Now you can turn your electric typewriter into a hard-copy printer with the I/O Pak from Rochester Data, Inc., 3100 Monroe Ave., Rochester, NY 14618. This typewriter interface exploits the high quality and full upper and lowercase characters of electric typewriters, permitting users of small computer systems to expand those systems into applications demanding high-quality text, such as word processing.

The I/O Pak, consisting of an array of coils positioned in the same pattern as the typewriter's keyboard, fits directly over the keyboard. These coils are wired into an electrical decoding matrix. The unit is designed to operate on voltages available from standard computers; no modification to the typewriter is required. All adjustments to compensate for different



The Tuscan S-100.



The HKB-1.

key heights are incorporated in the I/O Pak. Interfaces and software are available for the TRS-80, Level I and II, and the Apple II. A 6-bit parallel interface for general operation with other computers is also available. Price is $499. Reader Service number 479.

TRS-80/H14 Interface

Now you can interface your TRS-80 and the Heath H14 Serial Printer without having to load a software driver into memory each time the computer is powered up. The PTS-3 interface plugs into the parallel printer port of the Radio Shack Expansion Interface, and the H14 connects to the DB-25S connector of the PTS-3. The H14 baud rate switches are set for operation at 4800 baud. All handshaking and printer status signals are supported with this interface.

Once the PTS-3 is installed, the TRS-80 "thinks" it is connected to a Centronics-parallel-type printer. Compatibility is extended to support all printer commands whether at the BASIC level or machine-language level.

The PTS-4 interface can be used with the PTS-3 in systems that do not include the Radio Shack Expansion Interface. The PTS-4 simply connects to the 40-pin card edge located on the rear of the TRS-80 keyboard. The PTS-3 can then be connected to the PTS-4 to obtain printer operation, just as if an Expansion Interface were being used. The PTS-3 and PTS-4 each cost $69.95, plus $3.50, shipping and handling.

Multi Media Systems, PO Box 41084, Indianapolis, IN 46241. Reader Service number 478.

Britain's S-100 Microcomputer

The Tuscan is a newly designed Z-80 single board computer that is based on the S-100 bus. Billed as the first British S-100-based microcomputer, it utilizes widely available S-100 extension cards and comes with five S-100 cards laid flat on one board. It features versatile I/O capability with immediate expansion possibilities, including a disk-based CP/M system, high-resolution graphics and speech synthesis. Packaged in a professional case with integral disk drives, the unit is available in all options as a kit or fully assembled and tested.

Transam, 12 Chapel St., London, England NW1 5DH. Reader Service number 484.

H8 Prototype Board

The HKB-1 prototype board for the H8 bus, designed for ease of external cable connection, is a full-sized FR-4 board with heat sink/mounting brackets, bus connectors and a polarizing key. It uses .042 inch diameter plated-through holes on .1 inch centers for use with wire-wrap pins or direct solder connections. It features a hole pattern with interlaced power and ground traces with built-in jumper locations available. Price is $46, kit.

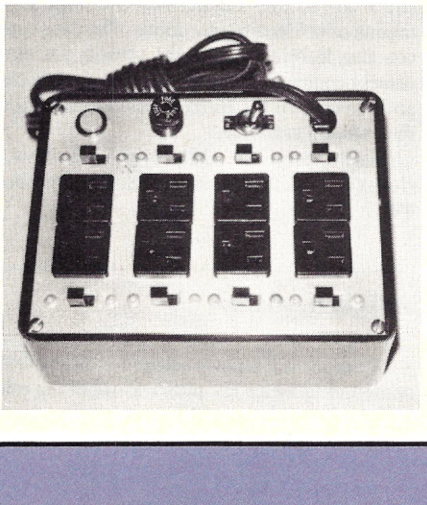
Mullen Computer Products, Inc., Box 6214, Hayward, CA 94545. Reader Service number 482.

Power Control Console

Spike-Spiker is a computer power control console that makes it convenient to plug all your computer equipment into one unit and switch the equipment on and off in the required sequential order. It eliminates constant plugging and unplugging of power cords. It also protects your computer from power line transients with an absorber and provides rf hash filtering between the computer and motorized equipment. The console has eight individually switched 120 V ac outlets divided into two separate filtered circuits, main on/off switch, fuse and indicator light. Price is $44.95.

Kalglo Electronics Company, Inc., Colony Drive Industrial Park, Box 2062, Bethlehem, PA 18001. Reader Service number 472.

Kalglo's Spike-Spiker.



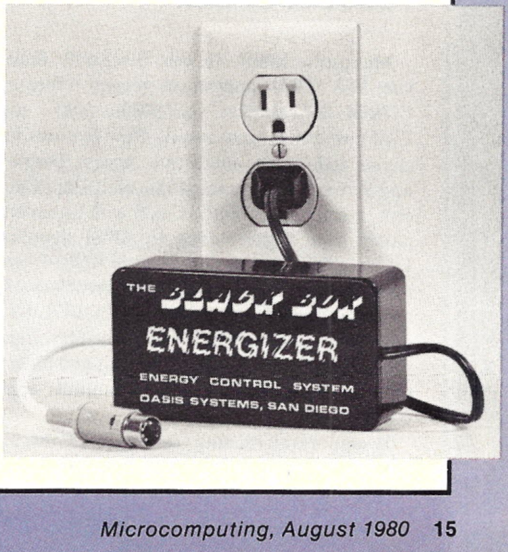
TRS-80 Power Control Interface

The Black Box Energizer plugs into any Level II TRS-80 to control up to 256 separate appliances and lamps. A built-in timer measures time from seconds to days, with 1/60 second accuracy. Lamps can be dimmed and brightened under full program control. An exclusive fast-control mode is provided for special applications such as lighting displays. It is also suitable for industrial applications such as automatic irrigation, solar energy, security systems and manufacturing control.

This power control interface works with any appliance or lamp control module manufactured by BSR and sold separately by Sears and Radio Shack. It broadcasts control signals directly over your home's electrical wiring. A complete system requires only the energizer and one or more power control modules (sold separately). Price is $49.95.

Oasis Systems, 2765 Reynard Way, San Diego, CA 92103. Reader Service number 474.

The Black Box Energizer from Oasis Systems.



Edited by Dennis Brisson

NEW SOFTWARE

Computer Bismarck

Computer Bismarck is a simulation game modeled after the confrontation between the British and the German naval units at the outset of World War II. It transforms the simple Battleship game into a game of advanced strategy and planning. The luck element has been minimized, and skill, cunning and planning are the critical factors. I received two versions of the program for review: Apple disk and TRS-80 cassette.

The Apple disk is more entertaining because of its use of color graphics and the quick access to disk data that allows for faster play between the different program elements. The program consists of the setup, main program and combat program. The disk uses a single data file for all three parts and has auto load linking between the program elements.

The TRS-80 cassette version contains both 16K and 32K Level II. The 16K version is cumbersome to use. The setup program produces a data file on tape that has to be read by the main program for play to begin. Before engaging in combat, you must make a data file tape which is read by the combat program. Due to memory restrictions, each program section has to be loaded separately and run individually. The 32K version has the main and combat program elements combined, so it will play much easier and quicker than the 16K version. I would like to see a 48K disk version for the TRS-80. Apple disk version is $59.95; TRS-80 cassette version is $49.95.

Strategic Simulations, Inc., 450 San Antonio Rd., Suite 62, Palo Alto, CA 94306. Reader Service number 489.

Ed Umlor
Technical Dept., ISI

COBOL-80 Compiler

Microsoft, 10800 NE 8th, Suite 819, Bellevue, WA 98004, announces version 4.0 of its COBOL-80 compiler for 8080-, 8085- and Z-80-based microcomputers. New features include: full-screen interactive Accept/Display and Screen Section compatible with Data General Interactive COBOL, Chain with argument passing and Segmentation to ANSI standard Level I. This new version exceeds ANSI-74 requirements with full implementation of Level I, as well as many Level II features. COBOL-80's advanced features—full Copy facility, trace style debugging and ASCII, packed and binary data formats—maximize microcomputer utility.

It supports all existing versions of CP/M, including 1.3, 1.4 and 2.X for files up to 8 megabytes. It runs under CP/M, ISIS-II, IMDOS, CDOS, TEI's TDOS and Model II TRSDOS operating systems; it can be easily adapted to other operating systems. Price is $750; documentation may be purchased separately for $20. Reader Service number 486.

TRS-80 Software

Simulation Software, PO Box 1368, Warren, MI 48090, announces the release of two programs for TRS-80s equipped with Level II BASIC and 16K RAM.

Dungeon Explorer 2.0 is a single-player game of combat and adventure in which a player tries to become a superhero by battling monsters within the Dungeon of Xanadu. This revised version features a streamlined game command input routine (using INKEY$), improved combat sequences, additional monsters and mapping graphics. No two trips into the dungeon are quite the same.

Cosmic Trader is a multiplayer game of interstellar trade. Up to four people try to amass a fortune by commanding their own star freighter in a quadrant consisting of nine star systems with nine categories of trade goods. Players must negotiate all transactions with alien merchants (the computer). Players must cope with sudden changes in the marketplace and in market prices. The user can adjust the game length.

Both programs are on cassette and come with complete instructions for $12.95, plus $1 per order for shipping. Reader Service number 487.

Nutritional Software

Nutri-Pack is a series of programs and a data base for the Apple II to help you evaluate the nutritional quality of your daily diet. The programs allow you to quickly retrieve information from, modify and add to a data base containing over 600 different foods. The data base contains information on the caloric, fat and protein content and the levels of eight vitamins and minerals in the listed foods. Price for the disk version is $39.95.

Micro-Comp, Inc., 2015 NW Circle Blvd., Corvallis, OR 97330. Reader Service number 490.

Adventure Game

Dungeons is a fantasy adventure for the OSI computer in which the player assumes the role of a fighter, dwarf, halfling, elf or magic-user in search of gold in the unexplored dungeons beneath the wizard's city or in the forest that surrounds the city. Evil monsters lurk in the forest and dungeons to guard the gold. It is based on the Dungeons and Dragons game. The adventure is graphically displayed for the C1, 2, 4 and 8P. Price is $12.95 for the cassette and $15.95 for 5¼ or 8 inch disk. Both versions require 8K.

Aurora Software Associates, 353 S. 100 E. #6, Springville, UT 84663. Reader Service number 488.

For Dentists Only

Graham-Dorian Software Systems, Inc., 211 N. Broadway, Wichita, KS 67202, introduces a computer software dental package written and tested by dental professionals. It handles patient records of charges, payments, insurance, delinquent accounts and daily and monthly transactions. It prints out patient statements and standard insurance forms for the American Dental Association (ADA).

The package can be ordered on standard 8 inch disk or various mini-floppy disks. Each package includes the software in INT and BAS file form plus a user's manual and hard-copy source listing for easy customizing. The package utilizes a two-disk storage system. Reader Service number 491.

Inventory Program

Micro Business World, 15818 Hawthorne Blvd., Lawndale, CA 90260, announces the Inventory Control System for the Apple II. The program will handle up to 8100 items and contains a transaction register, fast data retrieval and audit trails. It will generate inventory status reports, reorder reports and keep track of purchase orders automatically. It may be used in a retail or wholesale environment and will handle multiple departments or divisions. Minimum hardware requirements are an Apple II Plus with 48K, one disk drive and an 80-column printer. Price is $99. Reader Service number 492.

DBMS Business Program

Info/80 is a data base management system (DBMS) that runs under the CP/M operating system by Digital Research and utilizes Microsoft's Compiled (C-80) BASIC. This product features an effective and user-oriented method

of maintaining bookkeeping, recordkeeping and management information systems tailored for individual businesses. It is operational on various disk devices ranging from 8 inch diskette to multi-megabyte hard disk with the size of the data base the controlling parameter. Thus, it can manage both a small limited application and a complete multi-application for a full on-line integrated business system.

Data Train, Inc., 840 NW 6th St., Suite 3, Grants Pass, OR 97526. Reader Service number 493.

Business Software

L216 is a business software package for TRS-80 systems with 16K memory and Level II BASIC. It consists of the following programs: a cassette data base manager, a word processor, an inventory control system, a stock management program, a label printer, a deposit calculator and a statistics program. It also features a sort utility and a key access utility, which can be included as part of the user's program. Price is $59.

Micro Architect, Inc., 96 Dothan St., Arlington, MA 02174. Reader Service number 494.

OSI Compiler

XPLO is a block-structured, high-level compiler language for Ohio Scientific computers. This new programming language includes a self-contained editor and run-time interpreter. The editor allows easy source code creation and editing, and the interpreter makes XPLO programs transportable to any computer that has the interpreter written for it. Also, the block structure allows the creation of easy-to-understand, self-documenting code.

The diskette package ($79) comes complete with utility programs: DIRectory, CREATE and DELETE, all written in XPLO. The diskette also has some sample programs in XPLO. The cassette version costs $75. The 34-page manual, which may be purchased separately for $9.95, has sample programs, tips and a section on using the editor.

Pegasus Software, PO Box 10014, Honolulu, HI 96816. Reader Service number 495.

AppleRoots

AppleRoots is a combination genealogy/animal breeding program that has 17 user-definable fields to specify the title and length of the field. The program will default to 17 titled fields. Functions include: configure system, enter records, change records, delete records, print index or records, print list of children, print family records and print four-generation pedigree chart. All printer functions can be displayed on the screen or sent to the printer. All functions are menu-oriented; no programming is required to custom-configure the system for your personal use. It is written in Applesoft and requires a single disk drive with 24K RAM. Price is $39.95.

CDS Corp., 695 East 10th North, Logan, UT 84321. Reader Service number 498.

BASIC-FORTRAN Translator

Now you can convert software written for DEC, IBM and any other ANSI standard FORTRAN system into microcomputer-compatible BASIC with Convert, a software package for translating programs in BASIC to FORTRAN and programs in FORTRAN to BASIC.

Convert allows special BASIC command definition and FORTRAN device number specification to ensure accurate translation between any microcomputer and FORTRAN system. This translator is available in either the version I source code written in BASIC or the version II source code in FORTRAN. Both versions will operate on all popular computers with either a BASIC or FORTRAN compiler, respectively, and a minimum of 8K. The program is supplied on cassette for Ohio Scientific and TRS-80 and on floppies for Alpha Micro. A tape is also available for DEC, Prime or IBM systems. Price is $115.

Cognitive Electronics Laboratory, PO Box 615, New Braunfels, TX 78130. Reader Service number 497.

Pinball Game

Pinball is an arcade game written in machine language for the Radio Shack Model I Level II TRS-80. The screen displays flippers, bumpers, rollovers, runs and bonus points. The space bar on the TRS-80 releases the ball at various speeds under player control. Ball speed and acceleration depend on the contact with various features on the board, including the mysterious "Bermuda Square." Price is $14.95 on cassette or $20.95 on disk.

Acorn Software Products, Inc., 634 North Carolina Avenue, S.E., Washington, D.C. 20003. Reader Service number 496.

MDBS Software

Micro Data Base Systems, Inc., PO Box 248, Lafayette, IN 47902, has recently released version 1.03 of its network data base management system (MDBS) designed to run on Z-80, 8080, 8086 and 6502 processors. The current version requires about 18K on Z-80 machines. MDBS furnishes a collection of different, relatively simple commands. The task performed by each command is identified in the command's mnemonic. You are not restricted to getting, modifying and deleting data for the current record of the run-unit only. The design of MDBS allows you to write and execute application programs without having previously defined subschemas for them. The MDBS-Z-80 costs $750; prices for other processors are higher. Reader Service number 499.

NEW PUBLICATIONS

Challenger III Service Manual—handbook containing fold-out schematic diagrams, pictorial diagrams, block diagrams, parts lists, memory maps, board placement diagrams and component pin-outs for the 13 circuit boards used in OSI's business computer systems.
Ohio Scientific, 1333 Chillicothe Rd., Aurora, OH 44202.

MicroShopper 80: The New Computers—1980 edition of business and personal guide to microcomputer hardware and software, including over 500 products.
The Phoenix Group, 1425 West 12th Place, Tempe, AZ 85281.

Heathkit Spring Catalog—free 104-page catalog that contains descriptions of nearly 400 different electronic kits for home or business.
Heath Company, Dept. 350-270, Benton Harbor, MI 49022.

Micro Media Magazine—floppy-disk-based bimonthly publication that features software, reviews, graphic art, advertisements and articles for the Heath H8, H88 and H89 and the Zenith Z89. Available in both Benton Harbor and Microsoft, as well as in either HDOS or CP/M disk format.
Micro Media Magazine, 1316 Elmhurst Dr., Garland, TX 75041.

Computers in Psychiatry/Psychology—bimonthly newsletter for professionals interested in the use of computers in psychiatry and clinical psychology.
Computers in Psychiatry/Psychology, 26 Trumbull Street, New Haven, CT 06511.

Data Bits—monthly newsletter that coordinates nationwide the data and automation efforts of health planners within the 205 health systems agencies and 51 state health planning and development agencies.
Hapenny Associates, PO Box 1076, Columbia, MD 21044.

Software Vendor Directory—listing of over 700 vendors within 35 categories of hardware and operating systems.
Micro-Serve, Inc., PO Box 482, Nyack, NY 10960.

Archer Engineer's Notebook—handbook of 415 electronic circuits for electronics hobbyists, experimenters, technicians and engineers.
Radio Shack, 1300 One Tandy Center, Fort Worth, TX 76102.

Nibble—magazine published eight times a year that focuses on the Apple II and Apple II Plus computers.
Micro-Sparc, Inc., PO Box 325, Lincoln, MA 01773.

All About Personal Computers—report that traces the development of the personal computer, discusses applications and future trends and outlines how to buy a computer.
Datapro Research Corporation, 1805 Underwood Blvd., Delran, NJ 08075.

Robert W. Baker

PET-POURRI

Carl Moser of Winston-Salem, NC, was the first to provide a comprehensive machine language assembler for the PET and other 6502-based systems. The assembler program itself was written entirely in machine language and ran much faster than other assemblers written in BASIC.

The program was not very well publicized or advertised, since it required at least 16K of memory and was not practical on an 8K PET. But Moser has teamed up with J. R. Hall and formed Eastern House Software (EHS) of 3237 Linda Drive, Winston-Salem, NC 27106. They are now advertising several products for the 6502 microcomputer market, primarily for the PET.

Their most interesting product is the MAE, a new macro assembler/text editor package for the 32K PET with a 2040 disk drive. This package is similar to the older assembler, but is now greatly enhanced with the exclusive use of disk files. MAE appears to be the most powerful 6502 assembler and text editor package on the market.

The MAE comprises the macro assembler (ASSM) and text editor (TED), which reside simultaneously in 10K bytes of memory (5000–77FF hex). In addition to the 10K for MAE, sufficient memory must be allocated for label and text files, which normally take up locations 1800–2FFC and 3000–4FFC hex respectively. These boundaries leave memory for an extended monitor at 1000–17FF hex, for DOS support (wedge) in upper memory, and for BASIC and machine language programs at 0400–1000 hex.

The label and text files are position-independent and may be located almost anywhere in RAM memory. In addition, records within these files are variable in length and directly dependent on the number of characters to be stored. This results in more efficient memory use.

The TED occupies about one-half the memory space. It sets up and maintains the source file by interacting with the user via 27 commands (Table 1). When inputting to the TED, you have the full capabilities of the built-in cursor-oriented screen editor, and can automatically repeat any key held for a half-second or more. Source files are created and edited much like BASIC programs are normally handled by the PET operating system, so it is very easy to get used to.

ASSEMBLE:	Loads text file and begins assembly.
AUTO	Begin auto line numbering with next user entered line number.
ALPHA:	Select whether shifted characters will be graphics or lowercase.
BASIC:	Restore zero page and go to BASIC.
BREAK:	Restore zero page and go to monitor.
CLEAR:	Clear text file.
COPY:	Copy lines in the text file to a new location.
DC "command":	Pass disk command to PET 2040 disk.
DELETE:	Delete entries in text file.
EDIT:	Edit text file line or perform string search and replace.
FIND:	Search for string.
FORMAT:	Sets/clears formatting of text file when output. When formatted, fields of each line will be in tabular form.
GET:	Reads text file from disk into text buffer.
HARD:	Sets/clears formatting of listings for hard copy with page numbers.
LABELS:	Prints all or part of the label file.
MANUSCRIPTS:	Selects whether or not line numbers are printed. With this feature, MAE can be used as a simple text editor to create manuscripts or program documentation.
NUMBER:	Renumbers the text file lines.
OUTPUT:	Creates a relocatable object file on disk.
PASS:	Executes second assembly pass if first pass was previously performed.
PRINT:	Prints the text file data.
PUT:	Writes all or part of the text file to disk.
RUN:	Initiates execution of a previously assembled program. A symbolic label can be entered and MAE will find the correct starting address from the label file.
SET:	Reads or sets the text file, label file, and relocatable buffer boundaries. It can also be used to determine the remaining space in the text file.
TI:	Assigns terminal output to the PET or a serial device. Optionally, complete input and output can be assigned to an external serial device. MAE contains a built-in software UART routine to communicate with the serial device via the PET user port. Permissible baud rates range from 110 to 9600 baud.
TO:	Assigns terminal output to the PET CRT, IEEE device #4 (printer), or an external serial device. Optionally, output can be directed to both the IEEE and external serial device.
USER:	Restores zero page and goes to location $0000.

Note: Most of the commands have many options to allow ranges of lines, specifying strings and file names, etc. This list is only meant as an overall summary of the MAE TED capabilities, which are extensive.

Table 1. MAE TED commands.

The assembler scans the source program in the text file and creates a label file (or symbol table) on the first pass. An optional listing is generated during the second pass, and a relocatable object file can be generated by a third pass. The relocatable object file is recorded on disk and, with a separate relocating loader program, can be relocated almost anywhere in memory.

The loader can relocate your program in three segments: page zero variables, absolute variables and program body. When not generating relocatable object, the assembler can store the executable object code directly into memory. The code can even be stored at a different address from its execution address. This can be useful if you want to execute in memory space occupied by MAE or any of its work files.

The assembler source statements consist of a required line number, along with standard label, mnemonic, operand and comment fields, in a free format. That is, each field need not start in a specific column or character position. Labels can be up to 31 characters long while standard 6502 mnemonics and addressing mode formats are used. Symbolic, decimal, hex, binary and ASCII values can be entered and expressions can contain addition or subtraction operators. There is even a way to obtain just the high or low part of an address.

The conditional assembly features direct the assembler to conditionally assemble certain portions of your program and skip other portions. The macro facilities are extensive, with non-repeating labels, nested macros and conditional assembly within macros. Table 2 lists the standard assembler pseudo ops.

Source for a large program can be divided into several modules, each entered into the text file one at a time and recorded on disk. These modules can then be linked during assembly via a control file, which specifies the order the modules are assembled in. At assembly, the assembler will load and assemble each module until the entire program is done.

The MAE assembler also provides a unique interactive assembly mode. The assembler can print messages and/or accept keyboard input during the first pass of the assembly. This provides many possibilities, such as specifying the actual assembly start address when the assembly begins.

For program debug, an extended monitor is included in the MAE package. This program is a 2K extension of the PET monitor that occupies locations 1000–17FF hex. A BASIC program on the MAE diskette provides an interactive review of the extended monitor commands and instructions on how it can be relocated to another area of memory. The extended monitor provides commands for dis-

assembly, enable/undo stop key, fill memory, hunt memory, interrogate memory, quick trace, memory transfer, walk code and others.

In four weeks of testing, I found the MAE package to be well documented, and I had no problems learning to use the text editor and assembler. The assembler was fast and program debugging went quickly. The ability to assemble and debug a program with everything resident in memory greatly reduced normal program debugging time.

This package should be indispensable if you are doing any amount of machine language programming, but the price ($169.95) may be a little steep for a home system. The package includes an excellent 44-page manual, which has clear instructions and many examples. It even includes a sample program to help clear up several areas.

The programs are distributed on a 2040 compatible floppy diskette with the following files:

- DOS support (wedge),
- Extended monitor object code,
- Extended monitor instructions,
- MAE object code,
- Relocating loader object code,
- Relocating loader relocatable object code, (so you can relocate the relocating loader),
- Library of PET RAM locations for the 32K PET in a MAE source file,
- Notes pertaining to MAE and
- Example program (UART driver).

.BA	Begin assembly at the specified address.
.BY	Store bytes of data specified in hex, decimal, binary or ASCII strings.
.CE	Continue assembly if not one of three fatal errors.
.CT	Designates current contents of text buffer as a control file. This allows use of the .FI pseudo ops to link to other files. Only one control file may exist during each assembly.
.DE/.DI	Define an external/internal label and its corresponding value.
.DS	Define a block of storage (with undefined contents).
.EN	Indicates the end of the source program.
.EC/.ES	Disable/enable output of macro generated object code in source listings.
.EJ	Eject to the top of the next listing page.
.FI	Assemble the specified file before continuing with the following statement. This pseudo op is only allowed in a control file!
.IN	Output a question mark and accept exactly four hex digits as input which are assigned to a label and stored in the label file. Input will only occur on the first pass of the assembly.
.LC/.LS	Disable/enable source file listing during pass 2.
.MC	Store object code at specified address even though assembled for a different address.
,MD/.ME	Start/end macro definition.
.MG	Declares the entire contents of the text as a macro global file.
.OC/.OS	Disable/enable storing of object code in memory.
.PR	Output the specified text during pass 1. This is intended to be used in conjunction with the .IN pseudo op to request user input during assembly.
.RC/.RS	Provides a directive to the relocatable loader to stop/start relocation requirements and store code at the proper address.
.SE/.SI	Define an external/internal address constant.

Table 2. MAE assembler pseudo ops.

By the way, another interesting product advertised by EHS is their PET RABBIT program. This is a 2K machine language program that allows loading an 8K program from tape in just 38 seconds. It also provides a RAM memory test, a keyboard auto repeat feature and several other commands. It costs $29.95.

A number of versions are available for each machine, depending on the desired code location. Contact EHS for more information on this and other products.

COMPUTER CLINIC

I am the proud owner of an 8K PET with upgraded ROMs and a Betsi waiting to be brought on-line. I am an analyst-programmer for the Australian government, so my background is in programming. I am interested in corresponding via programs or letters with any interested persons.

David Jones
34 De Graaf St.
Holder Australian Capital Territory 2611
Australia

Does anyone out there recognize any of the following computer circuit cards? These are all hamfest specials bought to build a cheap computer. Any photocopies of condensed manuals or schematics would be appreciated. Will pay for the favor. I've included all identification I could find.

1. Data media 8080A CPU card. 2DAAA005, 9.36 MHz xtal, dated April 21, 1978, 2 2101. Two empty 2101 slots, about 50 TTL, two Z 8-pin sockets, all chips TI, 100-pin edge connector (not S-100). (Got two for $5 each.) Maybe goes to a Datamedia smart terminal.

2. Small unknown 6503 card. TCI-1 logo. 6503, 6530, 6532 chips, 3.579 xtal, 36-pin connector on one side, 38 pin connector on other side, paper tag says P/A model uP7-1, Rev. C C11-80139 on card. (Paid $25.) Maybe a video processor?

3. Big Univac memory card. ID numbers 7318-2-73 (1973?), 38-75, BE-3, Assy. 4161700-05. Seventy-two Intel 4915636 MOS, B 7720A chips (18 pin, 256×4 RAMs?), two 100-pin connectors on one side. 3534009-01 Rev. G 127 stamped on other edge connector (Paid $5.)

4. Small RAM cards. Told they go to "Accukeyer"; "memory board 1769-25" stamp. Twenty-four Intel C1101A 256×1 RAMs, 2K×3, 44-pin connector. (Got four for $2.50 each.)

Charles Gerbino
1831 Stanley Place
Falls Church, VA 22043

The Psychology Department at the University of North Carolina at Chapel Hill is currently in the planning stages of a computer based "lab" for undergraduates. We chose the Apple microcomputer and decided upon Pascal as the programming language. We would appreciate hearing of any applicable software that is available.

R. F. Genovese
Dept. of Psychology
The University of North Carolina at Chapel Hill
Davie Hall 013 A
Chapel Hill, NC 27514

I am willing, even anxious, to open my Tektronix 4051 computer. Can anyone supply references and product names on how to modify this system and/or adapt other 6800 products to it?

Dr. George E. Sinclair
1985 Devonshire Drive
Sierra Vista, AZ 85635

LETTERS TO THE EDITOR

Two comments on W. A. Harrison's article, "Programming Optimization Techniques" (May 1980)

For shame! The very first rule in optimizing the performance of a program is to fully understand the environment in which the program is to execute. This means a rather detailed understanding of object code produced by your compiler, or the functional characteristics of your interpreter, and a detailed understanding of the instruction set and instruction timings of your processor.

Mr. Harrison's remarks may have held true on the system with which he is working, but to take that kind of ivory-tower thought and spread it around as universal truth is certainly a disservice to computer users generally.

I imagine that Mr. Harrison was working with a compiler, if he tested his examples at all. I suspect that most of his remarks would be appropriate in that environment, but I am sure that 99 percent of your readers are using an interpreted BASIC, rather than a compiled BASIC.

Getting down to brass tacks, in Example 8, three ways of skinning a cat are shown, marked "inefficient," "more efficient" and "most efficient." I have implemented these three approaches in Listings 1, 2 and 3, respectively. The execution times (on a TRS-80 model II) were 65 seconds, 68 seconds and 72 seconds, respectively, *exactly the opposite* of what Mr. Harrison would lead you to believe.

Generally, with an interpreter, it is preferable to reference an initialized variable rather than a constant. The process of fetching the value of a known variable is quicker than evaluating the literal. The failure of the "most efficient" example is simply based on the fact that the overhead of interpreting an extra statement for the temporary variable costs far more than the subscripted references.

I have seen many instances where a young programmer will go to great pains to optimize the efficiency of a program he is coding, only to be shot down because he doesn't really understand what is happening at the next level down.

I welcome technical articles on programming techniques, but please try to improve the level of applicability to the real world as opposed to painting towers an ivory color.

Robert Snapp
President
Snapp, Inc.
Cincinnati, OH

```
1000 CLS
1010 CLEAR 1000
1020 DEFINT A-Z
1030 DIM C(1000)
1040 A$ = TIME$
1050 PRINT TIME$
1060 K = 750
1070 FOR I = 1 TO 10000
1080 M = C(K) + 8
1090 Z = C(K) / 2
1100 NEXT
1110 PRINT TIME$
1120 END
```

Listing 1.

```
1000 CLS
1010 CLEAR 1000
1020 DEFINT A-Z
1030 DIM C(1000)
1040 A$ = TIME$
1050 PRINT TIME$
1070 FOR I = 1 TO 10000
1080 M = C(750) + 8
1090 Z = C(750) / 2
1100 NEXT
1110 PRINT TIME$
1120 END
```

Listing 2.

```
1000 CLS
1010 CLEAR 1000
1020 DEFINT A-Z
1030 DIM C(1000)
1040 A$ = TIME$
1050 PRINT TIME$
1070 FOR I = 1 TO 10000
1075 T = C(750)
1080 M = T + 8
1090 Z = T / 2
1100 NEXT
1110 PRINT TIME$
1120 END
```

Listing 3.

"Programming Optimization Techniques" was well-written and accurate with one glaring exception. Harrison stated that "access (speed) can be increased by using a constant as a subscript rather than a variable (i.e., V(7) instead of V(K))."

On any computer with BASIC language, at least those which utilize interpreters rather than compilers, the opposite is true. A variable subscript results in significantly faster operation time than a constant. This is because it takes the computer longer to convert a constant into its binary floating-point equivalent than it takes it to look up and retrieve a variable value.

Examples of the time difference are displayed on the printout from our Commodore PET computer running routines using variable subscripts versus constants (Listing 4). I verified the same time savings with a similar routine on a Data General minicomputer with time-sharing BASIC. I've found one of the keys to swift program operation to be the liberal use of variables in any application where the value is referenced repeatedly.

Steven G. Spearman
Hastings, NE

```
THIS ROUTINE TOOK 27.0833334 SECONDS

10 REM  ** ARRAY ASSIGNMENT USING
20 REM  ** CONSTANT SUBSCRIPT:
40 S=TIME:REM S=STARTING TIME (JIFFIES)
50 FORI=1TO5000:V(7)=4:NEXT
60 S=TIME-S:REM S=TOTAL TIME
70 OPEN1,4:CMD1:REM ACTIVATE PRINTER
80 PRINT:PRINT:PRINT:PRINT"THIS ROUTINE
   TOOK";S/60;" SECONDS"
90 LIST

THIS ROUTINE TOOK 25.0333333 SECONDS

10 REM  ** ARRAY ASSIGNMENT USING
20 REM  ** VARIABLE SUBSCRIPT :
30 K=7:REM  VARIABLE ASSIGNMENT
40 S=TIME:REM S=STARTING TIME (JIFFIES)
50 FOR I=1TO5000:V(K)=4:NEXT
60 S=TIME-S:REM S=TOTAL TIME
70 OPEN1,4:CMD1:REM ACTIVATE PRINTER
80 PRINT:PRINT:PRINT:PRINT"THIS ROUTINE
   TOOK";S/60;" SECONDS"
90 LIST

THIS ROUTINE TOOK 23.3666667 SECONDS

10 REM  ** ARRAY ASSIGNMENT USING
20 REM  ** ALL VARIABLES :
30 K=7:L=5000:W=4:B=1
40 S=TIME:REM S=STARTING TIME (JIFFIES)
50 FORI=BTOL:V(K)=W:NEXT
60 S=TIME-S:REM S=TOTAL TIME
70 OPEN1,4:CMD1:REM ACTIVATE PRINTER
80 PRINT:PRINT:PRINT:PRINT"THIS ROUTINE
   TOOK";S/60;" SECONDS"
90 LIST
```

Listing 4.

Harrison replies to criticisms

I enjoyed reading Mr. Snapp's letter immensely. As to his first remark, he is correct! When attempting to facilitate *machine dependent* optimizations upon programs, you should be familiar both with your machine and your translator, be it a compiler, an interpreter or an undergraduate research assistant toggling 0's and 1's into the front panel of your machine. My purpose in writing "Programming Optimization Techniques" was to present a number of commonly used *machine independent* optimization techniques.

Mr. Snapp's remarks about compilers vs interpreters were noted, and I agree with him on

that point. As he has shown, the use of a constant as a subscript is not more efficient than using a variable. However, when using a compiler (there's a rumor afoot here at the "tower" that there actually is a compiler or two available for the micros), the use of a constant as a subscript can result in a substantial improvement in performance. This is because the address of the array element can be computed and inserted at compile time. Because of this, a reference to the array element at execution time would be similar to a reference to a scalar.

On the other hand, if you use a variable as a subscript, the machine must constantly (or perhaps *un*-constantly?) compute the location of the element at execution time. I realize that this is dependent upon the use of a compiler, yet if Mr. Snapp's arguments were to be touted as "universal truths" the readers with compilers would be misled in a similar manner.

As for Mr. Snapp's Listing 3, it is rather obvious why it ran slower than either of the others. The very first point that I attempted to make in "Programming Optimization Techniques" was that invariant calculations should be put *outside* the loop (see Example 1). Therefore, if Mr. Snapp were to move line 1075, I'm quite sure that he would notice a substantial improvement in performance. After all, 9.999 needless evaluations of T = C(750) can be time-consuming.

As for "ivory-tower" thought, I'm afraid that it would be a bit presumptuous of me to consider "Programming Optimization Techniques" an example of *ivorytower-think* (to borrow from Orwell). Please do not confuse theory with uselessness. Almost everything which has anything to do with computers was no more than a theoretical concept five, ten, 20 or 30 years ago.

For a survey of "ivory-tower" optimization, I suggest you see:

- Aho, A. V., and Ullman, J. D., *The Theory*

(continued on page 218)

```
        inefficient                      efficient
00100  FOR K = 1 TO 1000          00100  B = C + R
00110        B = C + R            00110  FOR K = 1 to 1000
00120        Q1 = Q(K)B           00120        Q1 = Q(K)/B
00100  NEXT K                     00130  NEXT K
```

Example 1.

```
TEST:
    PROCEDURE OPTIONS (MAIN);
    DCL X(100) FIXED BIN (15) INIT((100)10);
    DCL N FIXED BIN (15);
    DCL I FIXED BIN (15);
         DO I=1 TO 25000;
              N=10+X(75);                      CPU Time:  11 Seconds
         END;
    END TEST;

    LISTING PL/I - 1

TEST:
    PROCEDURE OPTIONS (MAIN);
    DCL X(100) FIXED BIN (15) INIT((100)10);
    DCL N FIXED BIN (15);
    DCL I FIXED BIN (15);
    DCL K FIXED BIN (15);
         K=75;
         DO I=1 TO 25000;
              N=10+X(K);                       CPU Time:  12 Seconds
         END;
    END TEST;

    LISTING PL/I - 2

TEST:
    PROCEDURE OPTIONS (MAIN);
    DCL X(100) FIXED BIN (15) INIT((100)10);
    DCL N FIXED BIN (15);
    DCL I FIXED BIN (15);
    DCL ITEMP FIXED BIN (15);
         ITEMP=X(75);
         DO I=1 TO 25000;
              N=10+ITEMP;                      CPU Time:  11 Seconds
         END;
    END TEST;
```

Listing 7.

```
    INTEGER I(100) /100*10/
    DO 100 J=1,25000
         N=10+I(75)                 CPU Time:  6 Seconds
100 CONTINUE
    STOP
    END

    LISTING WATFIV-1

    INTEGER I(100) /100*10/
    K=75
    DO 100 J=1,25000
         N=10+I(K)                  CPU Time:  6 Seconds
100 CONTINUE
    STOP
    END

    LISTING WATFIV-2

    INTEGER I(100) /100*10/
    ITEMP=I(75)
    DO 100 J=1,25000
         N=10+ITEMP                 CPU Time:  4 Seconds
100 CONTINUE
    STOP
    END

    LISTING WATFIV-3
```

Listing 5.

```
    INTEGER I(100) /100*10/
    DO 100 J=1,25000
         N=10+I(75)                 CPU Time:  3 Seconds
100 CONTINUE
    STOP
    END

    LISTING FORTRAN IV - 1

    INTEGER I(100) /100*10/
    K=75
    DO 100 J=1,25000
         N=10+I(K)                  CPU Time:  4 Seconds
100 CONTINUE
    STOP
    END

    LISTING FORTRAN IV - 2

    INTEGER I(100) /100*10/
    ITEMP=I(75)
    DO 100 J=1,25000
         N=10+ITEMP                 CPU Time: 3 Seconds
100 CONTINUE
    STOP
    END

    LISTING FORTRAN IV - 3
```

Listing 6.

EXATRON STRINGY FLOPPY
Owners Association Newsletter

Secretary Fred Waters

This month we are presenting a slightly different view of the Stringy Floppy. For months we have expounded the virtues of the "Stringy" and our Program Chairmen and users group organization. This month we are presenting a letter written by one of the program chairmen in response to a written inquiry by a potential "Stringy" owner. We present the letter in its' entirety and without editing.

- - - - - - - - - - - - -

Richard Harrison
Rt. 2 Drysdale
Warrenton, Virginia 22186

6 May 1980

Mr. David L. Johnson
4106 Montreal Avenue
Prince George, VA 23875

Dear Mr. Johnson,

First of all please excuse the long delay in responding to your letter. The demands of my regular job keep me on the go for long periods of time.

I will key my response to the items as listed in your letter:

1. The STRINGY will give you excellent results with the TRS-80 Mod I (with or without the expansion interface.) STRINGY models are available for use with S-100 configurations: again results will be excellent.

2. Sorry, but cannot comment on machine language text editors for the Poly-88 system. Highly recommend the Electric Pencil for the TRS-80 (definitely over the R/S Script). Yes the Electric Pencil can be used with the STRINGY; and the files are stored on wafers.

3. I have several systems in use for general computing and specific applications, but the STRINGY is used on the TRS-80 Mod I 48K system. The STRINGY has been in use for one year now and other than two MECHANICALLY faulty tapes there has NEVER been a bad load or save with the unit. There is no 'Foolery' associated with the STRINGY, and as such it is a real work-horse which is elegant in its operational simplicity yet ultra-reliable.

4. My main application for STRINGY is in support of my Communications Repair business, i.e., inventory, billing, work processing etc. But that is not the ACID test for the unit. . .my kids are. The kids use it daily for games, education and general computer exploration. None of the sensitivities normally associated with floppy disc apply to the STRINGY. Sir, I have dealt with many computer companies over the past five years and there are only two which I have not found fault with either their product or the company officials, they are: EXATRON and APPARAT. What this means to you is that you will receive courteous responses over the telephone and prompt written replies, when requested (your chances of being ripped-off are slim with EXATRON).

Mr. Johnson, I am taking the liberty of sending a copy of this to EXATRON so that they may answer your questions regarding Poly-88 application in more detail and also get you on their literature mailing list.

Priced like a casette; performs like a disk.

Best Regards and Happy Computing.

Sincerely,

Richard J. Harrison

P.S. This letter is prepared using the STRINGY and Electric Pencil. If you are a HAM, will discuss this further over the radio simetime. My call is N2JR.

This is only a sample of the positive mail we receive each week and to us it is the best possible way to guage the success that we are achieving through our efforts to provide a good product at a reasonable price with real product support.

With this issue, we are finally introducing the APPLE version of the Stringy Floppy. It has been a long wait but they are finally in production. Please see our ad elsewhere for more information or call our toll-free hot line. Watch this space next month for the introduction of the PET version.

INFORMATION & ORDERS

The ESF is assembled and tested at the factory, with a 30-day moneyback guarantee and a one year full warranty. For fastest delivery, phone in your credit card or COD order using the toll-free line below.

Base price for the TRS-80 ESF, $249.50 (ask about the Starter Kit); for the S-100 ESF, $289.50; Apple with single drive is $299.50, dual drive is $499.50.

Info packets at no charge; users manuals for the TRS-80 ESF are available for $3.00 for shipping.

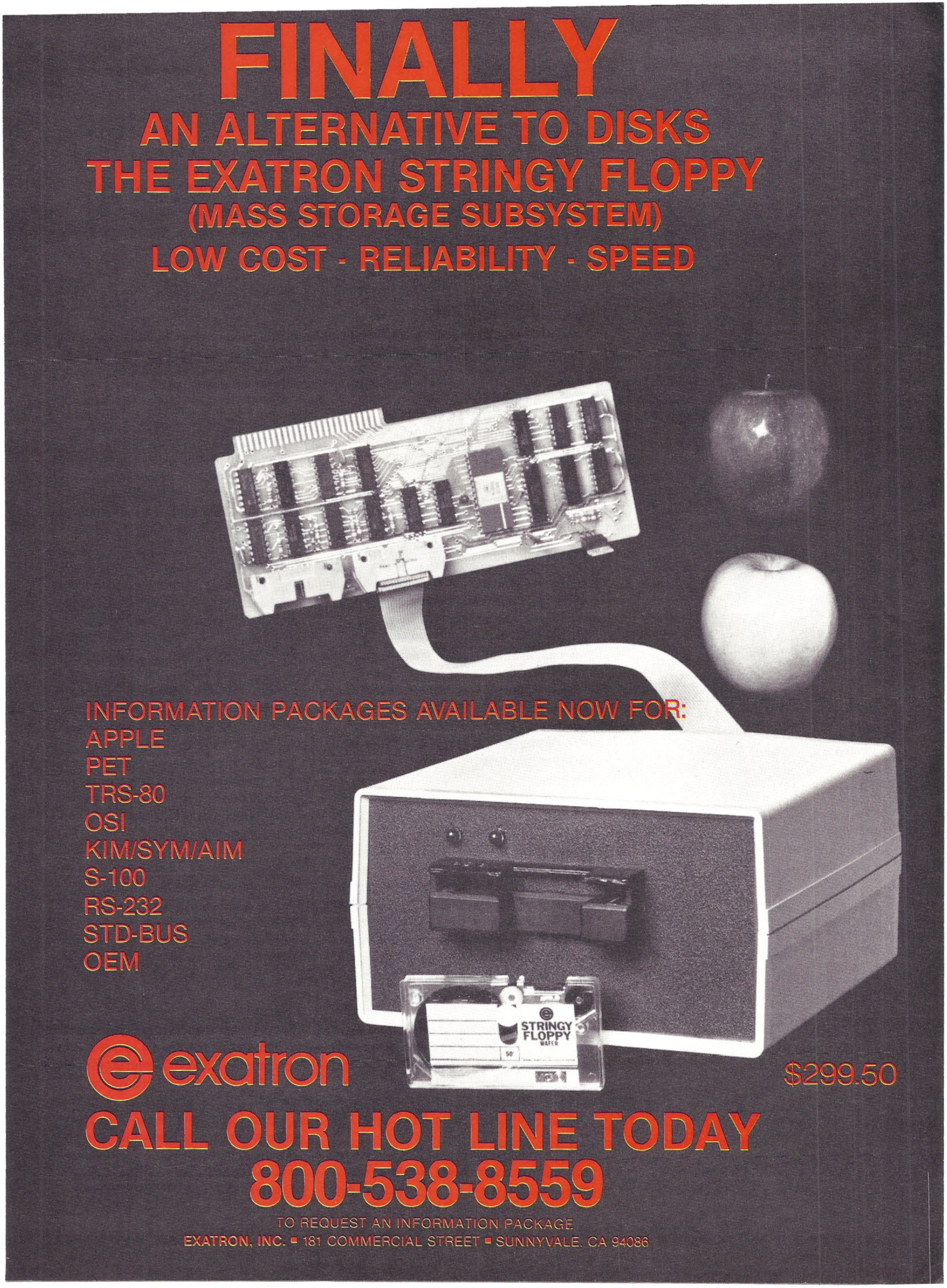
Handling is extra.

If you have any questions about these products, about Exatron or about ESFOA call the Hot Line. Address letters to ESFOA, 181 Commercial Street, Sunnyvale, CA 94086.

Stringy Floppy is a trademark of Exatron Corporation. ✓7

HOT LINE 800-538-8559

WITHIN CALIFORNIA 408-737-7111


FINALLY
AN ALTERNATIVE TO DISKS
THE EXATRON STRINGY FLOPPY
(MASS STORAGE SUBSYSTEM)
LOW COST - RELIABILITY - SPEED
INFORMATION PACKAGES AVAILABLE NOW FOR:
APPLE
PET
TRS-80
OSI
KIM/SYM/AIM
S-100
RS-232
STD-BUS
OEM
STRINGY FLOPPY
WAFER
exatron
$299.50
CALL OUR HOT LINE TODAY
800-538-8559
TO REQUEST AN INFORMATION PACKAGE
EXATRON, INC. ■ 181 COMMERCIAL STREET ■ SUNNYVALE, CA 94086

The 16-Bit Super Processors Are Here

This report zeroes in on the Intel 8086, the Zilog Z8000 and the Motorola MC68000.

Martin Moore
2735 S.W. 229
Beaverton, OR 97006

The gurus of the microprocessor business have been predicting a takeover by the 16-bit microprocessor for some time now. They told us about three years ago that the 16-bit microprocessor was just around the corner, and that the 8-bit machines had better heed the warning. And, as if to prove it, Texas Instruments came out with the TMS9900 microprocessor, the first, low-cost 16-bit microprocessor to hit the streets.

People didn't take to it immediately. The noise about 16-bit machines turned into a deafening silence. But it seems those gurus were right after all. They were just a little ahead of their time. The 16-bit revolution has come.

At this writing, at least four 16-bit microprocessors are available: Intel's 8086, Zilog's Z80000 (second-sourced by Advanced Micro Computers, Inc.), Motorola's MC68000 and Fairchild's 9440 Microflame.

The 16-bit microprocessors discussed here (I'm purposely leaving out the Fairchild 9440 because of its close resemblance to the Data General microNOVA) refer to memory as bytes, and for a very good reason. Few memory boards are set up to handle a 16-bit data bus. Almost everything available is for the 8-bit microprocessor. The manufacturers took this into consideration when designing their processors. Designing a microprocessor that can't access 8-bit memories is pointless. For this reason, all the manufacturers have built their devices to work with byte-oriented memory.

The 8086 Architecture

By using a silicon-gate HMOS process, Intel has managed to cram 29,000 transistors onto a very small die about 225 mm square (less than the area of Lincoln's head on the penny). That many transistors allow the 8086 to be divided into what are basically two processors. Fig. 1 shows a functional block diagram of the 8086.

Notice that the 8086 is divided into two halves called the bus interface unit (BIU) and the execution unit (EU). The 8086 performs a neat little trick by letting the BIU collect and send program information while the EU executes it. The BIU fetches instructions before they are required by the EU; it then loads the instructions in a stack, or queue, that will hold up to six bytes of instruction. The EU takes the instructions as it needs them, without having to wait for bus cycles, and without having to control such things as operand fetch and store, address location and bus control.

The process of parallel and simultaneous operation is called pipeline processing. It speeds instruction execution by never forcing the processor to wait while an instruction is being fetched.

Registers

The register structure of the 8086 is shown in Fig. 2. The 8086 contains 12 16-bit registers and a set of nine 1-bit flags. The asterisks in Fig. 2 represent the 8080A registers as a subset of the 8086 registers. This allows the 8086 to execute the 8080A instruction set without too much trouble.

The registers are grouped together within the 8086. Each group has a specific set of functions. The AX, BX, CX and DX registers are called the general register group, which are used in the arithmetic and logic operations of the 8086. Both halves of each register are separately addressable. Thus, you can think of this group as being two sets of four 8-bit registers.

The next four registers (SP, BP, SI and DI) are called the pointer and index register

group. They usually contain addressing offset values.

The instruction pointer (IP) register works in the same way as the 8080A program counter register.

The flag register contains nine 1-bit operation flags. The flags record 8086 status and are used to control the 8086 operation. Five of the flags are 8080A flags; four new flags have been added for the 8086.

Finally, the CS, DS, SS and ES registers, called the segment register file, are used in all memory address computations. For example, all instruction fetches are taken relative to the CS register, using the instruction pointer register (IP) as an offset.

These are not particularly general-purpose registers, and here is where the 8086 falls down in comparison to the Zilog Z8000 microprocessor.

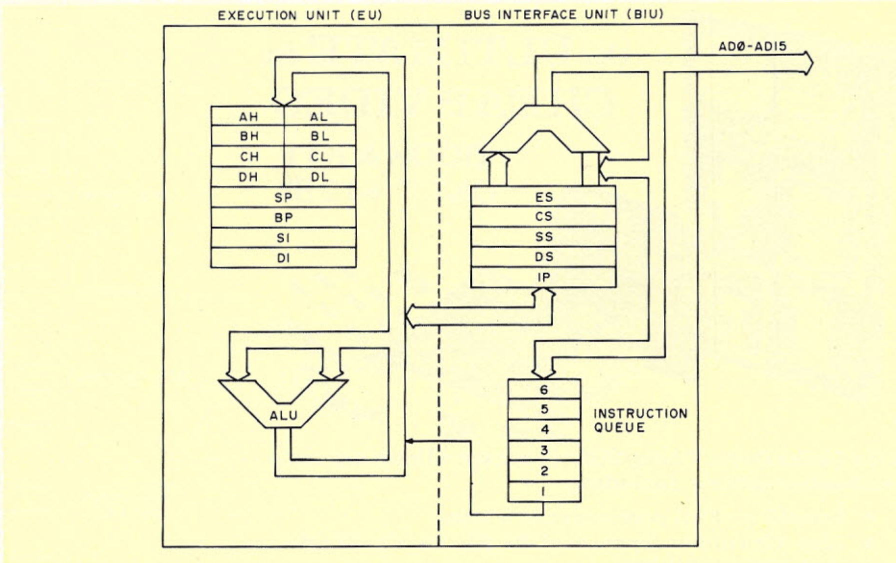


Fig. 1. 8086 architecture.

Memory

The Intel 8086 boasts a remarkable 20 address lines, which allow the 8086 to address over one megabyte (1,048,576 bytes) of address space. (That equals about 62.5 16K-byte memory boards.) Of the 20 address lines, 16 are time-multiplexed to act as the data bus.

For best performance, the memory would be arranged with the least significant byte of a 16-bit data word located at an even address, and the most significant byte at an odd address. That's the way the 8086 expects it. Instructions are fetched from memory as words; the bus interface unit (BIU in Fig. 1) loads the instructions into the queue as bytes for consumption by the EU.

In addition to the massive one megabyte of address space, the 8086 can be configured to address 64K 8-bit I/O ports.

The requirements for using this memory and I/O are very specific, and are best found in the Intel MCS-86 user's manual. Suffice it to say that memory space is not a problem with the 8086.

Instruction Set

The instruction set for the 8086 is divided into six functional groups: data transfer, arithmetic, logic, string manipulation, control transfer and processor control. Each of the first three functional groups is divided further into subgroups of instructions.

Data Transfer. Data transfer instructions are divided into four classes: general purpose, accumulator specific, address-object and flag. These instructions are used to move data to and from the 8086.

Arithmetic. The 8086 arithmetic instructions provide five basic mathematic operations: flag register settings, addition, subtraction, multiplication and division. These instructions have a number of varieties, including both 8- and 16-bit operations and signed and unsigned operations.

Logic. The 8086 provides basic logic operations for both 8-bit and 16-bit operands. In addition, it has three single-operand operations and four double-operand operations.

String Manipulation. The 8086 can manipulate byte or word strings with relative ease. The 8086's ability to repeat string operations within hardware is one convenient feature. Primitive one-byte instructions can be prefixed with a repeat number. Then, that instruction (or series of instructions) can be repeated n times, with no extra coding required. This feature can prove important when performing such jobs as code translation.

Control Transfer. The 8086 has four classes of transfer operations: calls, jumps and returns; conditional transfers; iteration control; and interrupts.

Processor Control. A variety of 8086 instructions control the processor. The 8086 can be halted, single-stepped and told to wait. Also, a one-byte prefix can precede any instruction to "lock out" any request to use the bus (as might occur in multiprocessor systems). During the time the 8086 is in this locked-out mode, all interrupts are masked. Interrupt requests are latched, but not acted upon until the lock prefix goes away.

Summary

The Intel 8086 was the first of the new generation of 16-bit microprocessors, and as such is probably better than the 8080A (if, indeed, you can compare the two).

Intel literature says that the 8086 can increase program execution speeds from seven to 12 times over the 8080A, while using 10 percent to 25 percent less code.

Also, the 8086's I/O capability should prove irresistible in any situation requiring an intelligent controller.

Z8000

Zilog's entry in the 16-bit microprocesser race is the Z8000. When I first looked over the Z8000 specifications, I was instantly reminded of that old workhorse of the minis, the LSI 11, from Digital Equipment Corporation.

The Z8000 (also produced by Advanced Micro Devices as the AmZ8000) is newer than the 8086 and shows it. The Z8000 has eight times the direct memory addressing capability (eight megabytes) of the 8086.

Architecture

The Z8000 is register-oriented. It has sixteen 16-bit registers, 15 of which are general-purpose, and an instruction set that supports over 400 combinations of instruction types, data elements and addressing modes. The Z8000 is simpler then the 8086 and resembles a minicomputer more than a micro.

Memory

Two versions of the Z8000 are available: a 40-pin version (called unsegmented) and a 48-pin version (segmented). The 48-pin de-

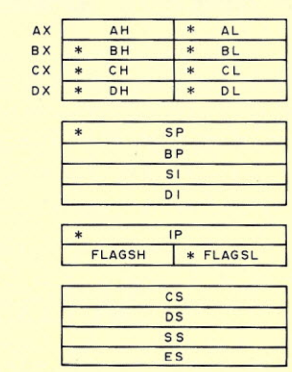


*Fig. 2. 8086 register set. *indicates 8080 registers.*

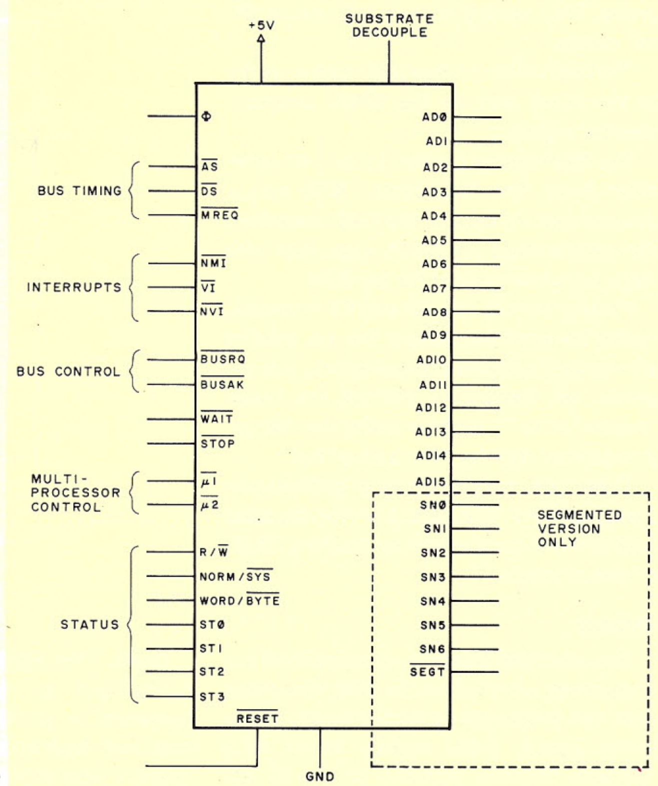


Fig. 3. Z8000 pin-out.

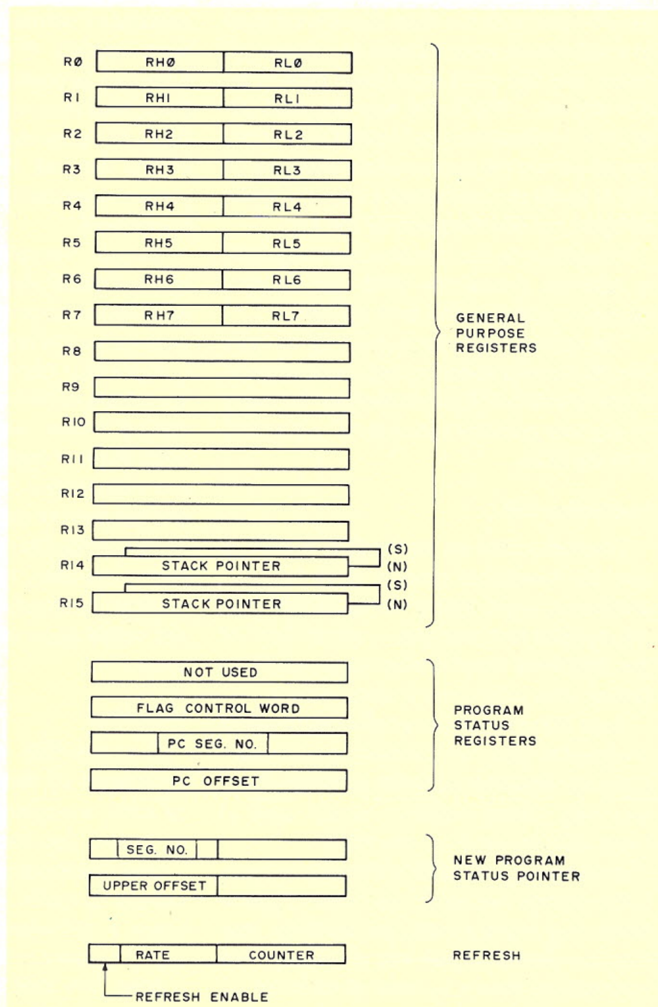


Fig. 4. Z8000 register set.

vice's extra eight pins increase its memory capabilities. I'll talk mostly about the 48-pin version. Remember that software written for the 40-pin version will run fine on the 48-pin version, but not vice versa.

The 48-pin Z8000 has the standard 16 address/data lines, much like the 8086 (see Fig. 3). In addition, the segmented version has an additional seven output pins (SN0–SN6 and SEGT) that extend the normal 16 address lines to 23 (SEGT is a control line). AD15 can address 64K addresses. If you use the segment lines SN0–SN6, you can point to 128 segments of 64K each. Using all 23 address lines, total memory is eight megabytes of address space (8,388,608 addresses, to be exact).

The basic data unit for the Z8000 is thought of as an 8-bit byte. It will as easily operate with 16- or 32-bit data words.

Back to the eight megabytes of address space. If, along with the Z8000, you purchase a memory management device, the memory capabilities of the Z8000 are multiplied sixfold. The memory management device (MMD) couples to the high-order address lines (AD8–AD15), the seven segment lines and the status lines (see Fig. 3). You end up with a chip set capable of directly accessing 48 million addresses. For all practical purposes, this is virtual memory. This all adds up in the following manner:

AD0–AD15 = 64K
SN0–SN6 = 64K × 128 = 8 megabytes
Status lines = 6 × 128 × 64K = 48 megabytes

The status lines serve to divide memory into system code space, normal code space, system data space, normal data space, system stack space and normal stack space. Each space can be addressed by a 23-bit address.

Aside from producing a huge address space, the MMD provides some other advantages. Consider the effort required to keep track of 48 million addresses. It is a monumental task at best, impossible at worst. Fortunately, the MMD, along with the Z8000 instruction set, does most of the work for you.

System/Normal. Recall that the status lines divide memory into system and normal code, data and stack space. The Z8000 can differentiate between your operating system code (system) and the code you're working on (normal). This allows the Z8000 to protect your system code from accidental alteration.

Addressing Structure. Addressing with the segmented version of the Z8000 is easy to do. The segment lines (SN0–SN6) establish a base address, pointing to one of 128 segments. The 16 AD0–AD15 lines point to a specific location within the addressed segment. The Z8000 registers are, therefore, designed to handle addresses in the same fashion.

Registers

The register structure of the Z8000 is one of its strong points. Fig. 4 shows the Z8000 register set. The 40-pin version has two stack pointers: a system stack pointer and a normal stack pointer. The 48-pin version has four stack pointers: 32 bits for normal operation and 32 bits for system operation. In the 48-pin version, 16 bits of the 32-bit stack contain the segment number, while the offset value is contained in the remaining 16 bits.

General-purpose registers. The Z8000 has 16 general-purpose registers, labeled R0 through R15. Each register can contain 16 bits of information. In addition, R0–R7 can each be divided into two 8-bit registers (RH0, RL0, RH1, RL1, etc.).

Word registers 32 bits long can be constructed from pairs of general-purpose registers (e.g., R0–R1, R2–R3). And 64-bit registers can be constructed from register quads (e.g., R0–R1–R2–R3, R4–R5–R6–R7). The 64-bit register quads are required by instructions such as Multiply and Divide.

Stack pointers. In the 40-pin non-segmented Z8000, R15 is doubled to act as the stack pointer (normal stack, system stack). In the 48-pin segmented Z8000, R14 and R15 are doubled to act as stack pointers (R14 contains the segment number, R15 the offset value).

Flag control word. The flag portion of the flag control word is the same for both versions of the Z8000. The flags used include carry, zero, sign, parity or overflow, decimal adjust and half-carry.

The control portion of the flag control word differs between the segmented and non-segmented versions. Control bits include vectored interrupt enable, non-vectored interrupt enable, stop mode, segmentation enable (48-pin version only) and system/normal mode.

Program counter register(s). The non-segmented version of the Z8000 has one 16-bit program counter register. The segmented version has two 16-bit program counter registers: the first holds the segment number, the second holds the 64K offset value.

New program status area pointer. This register contains the memory location of new program status words for the Z8000. In the non-segmented version, this register points to an address using only the upper eight bits of the 16-bit address bus. In the segmented version, two registers are used. One points to the segment number; the other points to the upper eight bits of the address.

When an interrupt or program trap occurs, the old program status words are pushed onto the system stack (identified by the system stack pointer register). New status words are fetched from the new program status area pointed to by the new program status area pointer register(s).

Refresh register. Refresh is a little less of a headache with the Z8000 than with the older microprocessors. A counter within the Z8000 automatically refreshes the dynamic memory. You can set a special memory refresh access at programmable intervals.

A programmable prescaler (a 6-bit modulo-n counter) is driven at one-fourth the system clock rate. The refresh register is nine bits wide and is automatically incremented by two each time the prescaler times out. This allows up to 256 rows of memory to be refreshed. The refresh feature can be disabled if necessary.

Instructions

While developing the Z8000, Zilog determined which instructions were used most often. Zilog then took these statically frequent instructions (the instructions most often found in a listing) and reduced the number of words required to execute them. Less code density was the result.

Some 110 distinct instruction types are used by the Z8000. Each instruction is divided into four (more in certain operations) basic fields. Those fields include mode field, indicating the addressing mode; the opcode field, indicating the instruction; the data element type, the byte or word designation; and register designation field, designating the register used in the instruction. An instruction can require from one to five words, depending upon its type and addressing mode.

Data types. The Z8000 can operate on five data types: BCD digits (four bits), bytes (eight bits), words (16 bits), long words (32 bits) and byte and word strings.

The byte is the basic data element. The number of bytes in any instruction is implied in the instruction, or in some cases is explicitly detailed by the programmer.

Bits, bytes and words (both 16 bit and 32 bit) are manipulated within the Z8000 registers. Byte and word strings, however, are stored in memory. String manipulation is eased by the use of the Z8000's auto increment/decrement addressing feature.

Instruction addressing modes. The Z8000 uses five main user-selectable addressing modes: register (R), indirect register (IR), direct address (DA), indexed (X) and immediate (IM).

In addition, several other modes are used for certain instructions: base address (BA), base indexed (BX), relative address (RA), auto increment and decrement.

Multi-processor capability. The Z8000, like the 8086 and MC68000, is specifically designed to work in a multi-microprocessor environment. The Z8000 has two pins included to ease multi-processor functions (see Fig. 3). The μ1 input disables the Z8000, while another processor is using a shared resource. The μ0 output lets the Z8000 prevent another processor from taking the bus if the Z8000 is using a critical shared resource.

Multi-processing is becoming more and more an economical prospect. The price drop in microprocessors versus their increasing power means that there is no reason to load one processor with all the work, when several can do the job better.

Summary

The Z8000 is, above all, a general-purpose machine. With its large addressing capabilities, its most common use will probably be in mainframe minicomputers, competing with DEC. The Z8000 is an advanced machine, but don't let that overwhelm you. The Z8000, like the other 16-bit processors, isn't any more difficult to understand than your 8080 or 6800. It's just a little bigger.

If you are interested in getting the complete story on the Z8000, get in touch with your local Zilog or Advanced Micro Devices representative.

MC68000

Now we come to the mystery machine. During its development, Motorola kept a tight lid on the MC68000. Now they're ready to send out small quantities for evaluation, though you still can't order more than ten. Motorola is rationing the MC68000 for fear that a few big buyers will snatch up 90 percent of their stock for the next year.

Architecture

The MC68000 appears to have been worth the wait. By using a HMOS fabrication method, Motorola has managed to put over 68,000 active devices on their wafer, as opposed to the 8086's 29,000.

The MC68000 comes in a 64-pin package like the T.I. TMS9900. Unlike either the 8086 or Z8000, the MC68000 has separate address and data pins. But like the 8086, the MC68000 is a pipeline processor. Recall that the 8086 fetches instructions before it actually needs them. The same thing happens in the MC68000. One half of the device performs instruction fetches, while the other half executes the instructions. Here again is an example of pipeline processing, something you'll be seeing a lot more of.

Registers

You will find sixteen 32-bit registers in the MC68000 (Fig. 5). They are divided into two groups: eight registers for data and eight for address.

The data registers can be used for byte, word or long-word data operations. The address registers are typically used for stack pointers and base address registers. In certain situations, the address registers can

INTRODUCING THE EMULATOR™

The trouble with video terminals today is that most of the low-cost models just don't have the performance to handle your tough applications. And the few that do are usually not compatible with your existing system. But now, Intertec has resolved this age old dilemma with the introduction of its new Emulator™ Video Terminal.

The $895* Emulator™ performs exactly as you command. With the depression of just a few keys, Emulator users can select terminal control codes of any one of four popular video terminals. The Lear-Siegler ADM-3A. The Soroc 1Q-120. The DEC VT-52. Or the Hazeltine 1500. Incredible! It's like having four terminals for the price of one.

But, best of all, not only does the Emulator replace these terminals, it outperforms them by offering enhanced user-oriented features. Features that those other terminals just don't have - at any price.

Standard Emulator™ features include: a sharp, crisp 12'' non-glare screen with a full 24 line by 80 column display. Twin RS232C serial ports - one for the host computer and one for your printer. Four separate cursor control keys. A separate 18 key numeric pad. Keyboard selectable baud rates and operating modes. And, a host of visual attributes.

No matter which dumb or smart terminal you're using today, don't buy another until you check out our new Emulator™. You'll get the performance of four terminals for the price of one. And you'll probably save hundreds of dollars over the price you paid for your last terminal. Plus, you'll get unparalleled reliability, nationwide service and quick delivery. Call or write us today for all the details. Intertec terminals are distributed worldwide and may be available in your area now.

INTERTEC DATA SYSTEMS®

343

2300 Broad River Rd, Columbia, SC 29210
(803) 798-9100 TWX: 810-666-2115

*Quantity one - Dealer inquiries invited.

also serve as word and long-word data registers.

As a side note, some believe that the MC68000, with its 32-bit registers, will act as Motorola's bridging processor into the 32-bit mini field. We'll have to wait and see. Notice in Fig. 5 that the eighth address register is actually doubled. The A7 register acts as the stack pointer for the MC68000. And the stack pointer is doubled, as in the Z8000, into a user (normal in the Z8000) and supervisory (system in the Z8000) register.

The program counter register is 24 bits wide, allowing the MC68000 a memory addressing range of 16 megabytes (16,777,216 addressable locations).

The status register has some interesting features, too. This 16-bit-wide register is divided in half. The user byte contains the normal status information you would expect to see. The system byte contains three bits for the interrupt mask, one bit to indicate a user or supervisory operating mode and one bit to indicate a trace mode.

Trace. The trace feature is unique to the MC68000. When the trace mode bit is set, the MC68000 traps to a tracing routine (that you write) after each instruction is executed. This valuable tool is like a built-in debugging feature. You can use trace whether you're operating in the user or supervisory mode, but you can only enter trace mode from the supervisory.

According to Motorola, the unused bits in the status register are for future expansion. Maybe they'll convert the MC68000 into a 32-bit machine.

Memory

The MC68000 has 23 address lines, labeled A1–A23 in Fig. 6. Notice that the MC68000 doesn't have an address line A0; the data bus is controlled in a byte-oriented fashion. That is, there are two control lines on the MC68000 called upper data strobe (UDS) and lower data strobe (LDS). These two lines remove the need to ever use the least significant bit of the address bus.

Motorola plans on making available a memory management controller that will handle memory segmentation and protection, much like the Z8000's memory management device.

Instructions

Motorola has taken a number of steps to enhance the MC68000 instruction set.

First, the MC68000 instruction set is a super-set of the old MC6800 instruction set. This was done to ease translation of 6800 code to MC68000 code. A translator will be available to perform this upgrading task.

Second, Motorola performed the same research Zilog did to look at statically frequent instructions (those instructions that occur most often in a program listing). But Motorola went a step further and looked for dynamically frequent instructions (those instructions that are most often executed). Keeping the numbers in mind, Motorola tried to create instructions that were as short as possible.

Third, Motorola prepared for the emergence of modularized high-level languages, such as Pascal. Several specific instructions in the MC68000 instruction set are geared directly for structured languages such as Pascal. Instructions such as LINK and UNLINK allow linked data lists to be manipulated within the stack areas. There are other examples, but the point is that the MC68000 hardware is set up to use structured languages efficiently.

There are 59 distinct instruction types in the MC68000 instruction set. Each instruction, with a few exceptions, will operate on data bytes, words and long words. And most instructions use any of 15 main addressing modes. If you combine the instruction types, possible addressing modes and data types, you end up with about 1000 distinct instructions.

And yet, with all those possible instruction combinations, the basic instruction list is easy to remember. If you can program a 6800 without having to look at the book all the time, you can probably program an MC68000 without looking.

As with the other 16-bit microprocessors we've discussed, the MC68000 will perform signed and unsigned multiply and divide operations in hardware, thus speeding arithmetic program execution. The microprocessor can deal with BCD arithmetic, as well as standard binary integers. The new MOVE data instruction will allow you to transfer bytes, words and long words in all data addressing modes.

Speaking of addressing modes, the MC68000 has five basic types: register direct, register indirect, absolute, immediate and program counter relative.

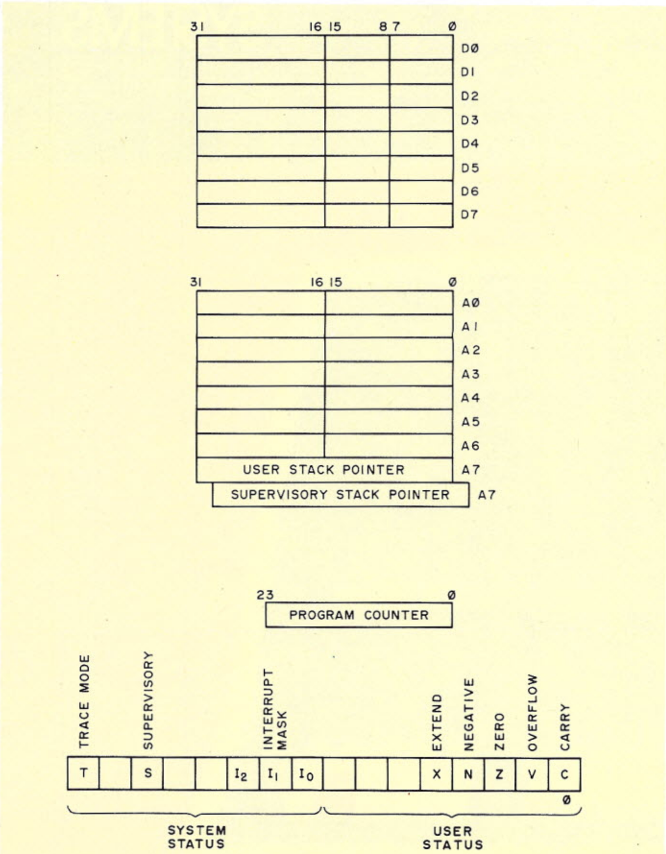


Fig. 5. MC68000 register set.

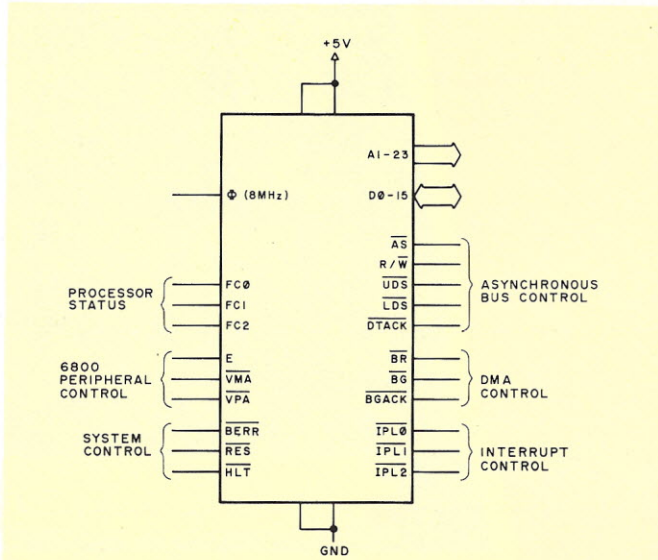


Fig. 6. MC68000 pin-out.

In the register indirect addressing mode, the MC68000 has two sub-modes called post-increment and pre-decrement. Here again, as in the Z8000, the MC68000 has enhanced data string manipulation capabilities. Overall, there are 15 addressing modes of operation.

Three pieces of information are required in an MC68000 instruction: the location of the operand(s), the size of the operand (byte, word, long word) and the function to be performed.

The MC68000 will operate with dual operands. The location of the operand in memory is either explicitly specified in the instruction or implied by the instruction as addressing modes.

This short explanation of the MC68000 instruction set does not do justice to the wide range of instruction possibilities, and I suggest that you seek out more information from Motorola if you're interested in the MC68000.

Support

The 8080, Z-80, and 6800 have a lot of support devices. Rather than design totally new devices for the MC68000, Motorola decided to implement the existing chips. Almost all of the 6800 family of peripheral devices can be used (in pairs, usually) with the MC68000. The VMA, E, R/W and RESET lines on the MC68000 are used exactly as they are on the 8-bit 6800. This means that you don't have to learn new peripherals as well as a new microprocessor when you use the MC68000. As far as I'm concerned, this is a big plus for Motorola.

Summary

I'm short-changing you in this brief outline of the MC68000. It is a remarkable device, and Motorola has apparently put a lot of forethought into its design. Literature describing this microprocessor in more detail will be available soon.

At this writing, Motorola is experiencing difficulty with the design of a buffer register within the MC68000. It doesn't want to pass data accurately. This will undoubtedly be cleaned up before the device gets into mass production.

What Does All This Mean to You?

First, does this mean 16-bit microprocessors will replace our reliable 8-bit processors? No. The 8-bit machines are too well entrenched to be dislodged by an increase in data length. After all, some 4-bit microprocessors are still around, used in simple control applications.

A plan is now before IEEE to adopt the S-100 bus, with provisions for a 16-bit-wide data bus. If this goes through (as it probably will), then S-100 will become standardized and will be capable of handling the new 16-bit processors.

The current outlook is that these microprocessors will be used in mini-type applications. They require massive memories to take advantage of their architecture. The part-time hobbyist may not want to become involved with the 16-bit processors. After all, purchasing 48 megabytes of memory is costly.

But if you're interested in plain old number crunching, with the maximum possible throughput, then you should definitely investigate this new breed of microprocessor. Speed is increased merely by doubling the data size for each instruction used. Other enhancements are included in the 16-bit machines, too. Long-word multiplication and division in hardware certainly won't hurt anyone's feelings.

Should you look into the 16-bit microprocessor? You bet! But if you decide to start implementing 16 bits, don't treat them any differently than your 6800. Use the expertise you've developed with 8-bit machines, and don't let the whiz-bang numbers fool you.

We're entering a new era in microprocessors (we seem to do this about once every three years). This will probably be a short era too. Next? 32 bits. In the meantime, let's sit back and watch the 16-bit revolution. ■

Micro-Matrimony

Can a law firm find happiness with a Sol 20 computer system?

Frank J. Derfler, Jr.
PO Box 691
Herndon, VA 36117

The marriage between the law offices of Piel and Lynn in Montgomery, Ala., and their Sol 20 computer system is one that has worked.

For less than $12,000, the firm has an electronic helpmate that does collection letters, divorce papers, incorporation papers and bylaws. It processes real estate closings in half the time it used to, and has reduced the time needed to prepare a will with a trust from two or three hours to 25 minutes.

And, says attorney Richard Piel, the system paid for itself in the first few months of operation.

Use of the Sol 20 revolves around the Word Wizard, a word-processing program from Processor Technology. Since paper is the major physical output of a law office, the computer was put right on the production line.

The disk files are loaded with documents and standard forms. The correct form is called from the file, the CRT fills in the blanks, and a Diablo printer produces the document. The result: a customized document that looks like an original.

Entering each document the first time takes a lot of work, and system operator Glenda Senn uses numerous control codes. She has taken one data processing course, and the rest of her training is on the job.

Buying the microcomputer was a family effort involving much research, says Piel. The firm spent two years looking for a system that would meet its needs economically. Many systems were available, but most dedicated word processors were expensive and rigid. They often controlled, rather than helped, office procedures because of a feeling that maximum use should be made of the expensive equipment.

Piel and Lynn do not do the actual number crunching of figuring accounts on the Sol. The system is used to address envelopes and print the bills. Piel is convinced that timely billing is the key to prompt payment, a viewpoint many businesses can appreciate.

The firm does not do accounts receivable or office typing on the system; much work is still done on standard office machines.

The key to the firm's success, says Piel, is that the company knew what it wanted. While local free-lance software people are looking at more uses for the system, anything the firm buys will have a fixed price and a specific purpose. The firm, says Piel, bought the computer system with realistic expectations, and was not driven by a need to get a return on a large investment.

Piel and Lynn's successful use of a relatively low-cost microcomputer system has several lessons for other small businesses thinking about a similar move.

- First, determine exactly what you want the system to do.
- Then, shop around. Do not be lured by pretty hardware.
- Buy general programs for a predetermined price.
- Put in some time setting up the proper forms.
- Do not let the system dictate the way you do things.
- Finally, be happy if you meet your original goals.

As in all marriages, realistic expectations and formal understandings of roles and responsibilities are necessary. It probably will not always be a bed of roses, but a warm relationship with a microcomputer can lead to unexpected delights. ■

This Sol 20 system found a happy home with a law firm. It greatly shortened the time needed to prepare legal documents and met all initial expectations.

Winning the Micro Game

Dilettantism won't do when it comes to learning about microcomputers.

Don Lancaster
Synergetics

Hands on is everything. The only way to ever learn anything about computers is to jump in with both hands and feet, get on line and do some computing. Until you actually do and see what the micro world is about, you've accomplished nothing. You must do things yourself, on your own terms, in front of a working, real computer, alone.

It's both funny and sort of sad to hear a student say he just took a DP course but couldn't get any CPU time. *He* got taken, not the course.

You become computer literate by using computers, not by having someone tell you about them or by reading about them.

Understand a timing loop by writing one and watching it work. Do an interface by taking a triac, an optocoupler and a 100-Watt light bulb and shining light on the real world. Find out what an interrupt is by interrupting a computer. Do it—yourself.

You have to make mistakes. If you are learning micros or developing any new product, half your experiments should fail. A canned set of exercises on a micro trainer is next to worthless if everything falls into place and works perfectly the first time.

In the micro world, you make mistakes to learn and to progress. You should expect mistakes. Prepare for them. Welcome them. Aggressively seek them out.

Of course, it makes sense to never make the same mistake twice. Build on what you have. To expand your microcomputer universe, try new things that may fail. Find out why they fail, and use this as a newer and bigger base to work from.

Usually, you are never anywhere near where you think you are in solving any hardware or software problem. Unexpected surprises and plain old stupidity are always between you and reality. If you think you have something working perfectly, you probably don't even understand the problem.

You must mix hardware and software. Some heads-in-the-clouds pure software people out there still believe that hardware is a mundane inconvenience standing between them and pure "computing." And there are technician types who do everything with bushel baskets full of integrated circuits.

Neither approach is good. Sometimes a simple and inexpensive hardware circuit can replace bunches of software. Other times and other places, a few lines of elegant software can eliminate the need for custom circuits or a special device.

Winning computer products will combine both hardware and software, using the best features of each to give you the simplest system and the lowest possible cost.

This means that if you are a hardware person, you should learn programming and learn it fast. If you have a software background, start soldering and wire-wrapping with a vengeance.

Synergy says that $1+1=4$. This is definitely the case when you get an optimum mix of hardware and software interacting with each other.

Neither can stand alone—not any longer.

The real world is fuzzy. Some textbooks and lab experiments work every time. Everything is nice and clean, neatly tied up. You do exactly what you need to do the job, no more, no less. Unfortunately, reality doesn't work that way.

First, you must deal with people, and that will always mess things up. Key items will be missing or late. The magic chip may be a figment of an ad writer's dreams. Or a problem may have a simple and inexpensive technical fix that is politically or socially unacceptable. Goals conflict. So do egos.

Expect and accept fuzziness. As you get into a new computer area, things will start out completely confusing. Then they will become fuzzy. Then they will become, for a glorious instant, crystal clear. Then, of course, they get fuzzy again as you become more involved.

As you go to the bigger picture, expect more fuzziness. Also recognize that there really isn't much in the way of real-world beginnings and endings. Rather, things sort of dribble off into the great whatever.

Micros might—just might—be the missing link between people and intelligent life in the universe.

Hit the basics hard. Any 6502 micro freak can sit down and immediately "prove" that the 6502 is ten times better than any other micro in the world. The trouble is that you can do the same with any other micro family, as well.

For most micro uses, it makes no difference which micro from which family you use. Even if there temporarily was a "best" micro, other factors such as your own skills and attitude, the available software, the elegance of your competitors' programs and so on will reduce any advantage of the "best" micro to zilch.

If you don't happen to like the "best" micro, just wait a month or two, and it will get shot out of the saddle by something much more promising.

This all means that the micro you learn is not the micro you will use. Later on, there will be much better ones to work with, and they are sure to have completely different tech details.

To beat this, hit the basics hard. All known micros have address space and addressing modes. All have interrupts, subroutines, clocks, ports, memory and I/O. Use any micro you like to add tech details to the fundamentals. But get the essentials down solidly.

Reach out and put the touch on someone. The nickels in the micro world are now to be made in places where people are not yet using micros. Find these places and get involved with these areas and people on their own terms.

Put micros to work feeding cattle, treating sewage, gambling on Wall Street, designing looms, mixing cement, baking calzones, milking goats, hulling pecans, questing tinajas, animating video, co-oping groceries, hybridizing sinsemilla, improving wood stoves, redesigning bicycles, restoring steam calliopes, monitoring steam gauges, selling paper clips, cutting dress patterns and teaching trumpets.

When you do reach out, always work in the other person's terms and language, bending the micro info to fit as you can. If they are smart enough to learn micros, they won't need you for anything.

Find places where they don't yet know that micros can help. Then jump in.

Don't reinvent the wheel—steal the plans instead. Much of the needed and obvious micro-related information has already been done and is readily available for your use. For instance, if you want to drive a Teletype or another printer, use someone else's driver routine. Don't stop what you are doing and invent your own—unless you truly want to know how a driver program works.

Scads of Morse code trainer programs are out there. Why write another? The same goes for sorts and word justification subroutines. And there are more versions of Lunar Lander than there are moons in the solar system. How many Hangman, Hexpawn or NIM games have you seen?

Now, if you want to learn these programs, that's fine. but, if your goal is using something, rather than creating something, find out what has already been done and go with it, or improve it and then go with it. Refer to monitor listings, user software exchanges, micro magazines, application notes, club newsletters, program books and micro information exchanges for programs to use.

Better wrong now than right later. In anything you do in the micro world, your first attempt will be wrong and will have to be reworked. So, immediately kludge up your first attempt and let your mistakes show you the way to go. Often you don't even understand what the real problem is until you are inside a program or a wirewrap board looking at it.

Try a simple, quick and dirty tactic that at least sends you in roughly the direction you want to go. Make some guesses. Take a stab at it.

In your early attempts, if it works, use it. Start your project flying more or less rightside up. Later on, you can go back and add structure to your programs, elegance to your methods, convenience to your user and simplicity to your hardware.

Add the final spit-and-polish on the way out the door, and not early in the game.

Write it down. And not on the back of an old envelope, either. Documentation is the password to avoid self-destruct modes. You don't record only final programs and schematics. Instead, keep track of what you did and why you did it, neatly, and in some semblance of order.

Keep accurate records of where you have gone and where you are heading. Put together the files of literature, instructions, op codes and program bits and pieces.

Software is worthless if you can't show someone else how to use it. Hardware has no way to operate if there's no way to connect it or fix it. You must be able to go back and reuse or modify what you did a week or a year ago.

Documentation is not just a hex dump; it's a micro way of life. You cannot survive without it.

Don't separate work from play. Which of these is more important: Designing an efficient sort algorithm for a business general ledger program or figuring out what to do with the oily slime in Adventure?

In the long run, the oily slime is far more important because that's what is stimulating interest in micros and making people computer-literate.

Any program run on a computer is a game! However, stuffy institutions, banks, bureaucrats and other so-called "serious" computer users have rules that say you aren't supposed to smile while you are playing their games.

Simon says don't smile. It's still a game. Have fun.

You will never get enough. No matter how far you have gone in microcomputing and no matter how much of what kind of hardware and software you have on hand, you will always "need" more of something.

More memory? Start with a 1K trainer, then 4K, then a 16K micro. Then overfill the 16 megawords of an extended micro space. Need hard copy? Start with Excedrin headache number ASR-33, then on to thermal, a Selectric and finally a daisywheel.

Now, if only daisy was intelligent and had its internal word processing.

From plain-jane video, go on to graphics, color graphics, hi-res and then super-resolution color with gray scale. From cassettes, it's on to floppy, dual floppies, quad density and then a Winchester.

> **"You can learn far more about micros watching fourth graders zap Klingons than you ever will in a university COBOL course."**

There never is, nor will there be, a time when you have "enough" of anything. What looks like a light at the end of the tunnel is a train speeding towards you.

You will find only one way out of the "more" syndrome. Always go with what you have. Make it work. Live with it as long as you can. Force it to pay its own way.

Make it do. Use it up. Wear it out.

If it's old line, stomp on it. Some premicro people and institutions are still kicking around the lunatic fringe of the new micro world. They persist with large, bureaucratic, centralized, insanely priced and unavailable megacomputers run by an elite priesthood singing the incantations of an arcane language. They completely fail to recognize the power of the micro as a highly personal, one-to-one, decentralized, inexpensive, interactive and individual convivial tool.

You can learn far more about micros in twenty minutes watching a pair of fourth graders zap Klingons than you ever will in a university COBOL course.

Old-line conventions include IBM, batch processing, COBOL, decollators, Honeywell, key-to-disk, FORTRAN, keypunches, centralized billing and data encryptation. They are without any redeeming social value. They had their chance and blew it. We

tried it their way and it didn't work.

Old line not only fails to see the problem; they are the problem.

Always ask, "Why are you telling me this?" The useful products and ideas in the micro world are not heavily advertised. In fact, anything genuinely useful takes a lot of time and trouble to nail down.

If a micro is widely or heavily advertised, it more than likely means that something much better is available elsewhere. If someone is radically trying to convert you to his microprocessor or his way of doing things, the chances are he has drifted into right field and become snookered into a bad scene. He is looking for converts to ease the pain when he is shot out of the saddle.

When anyone tries to tell you about micros, always ask, "What is the real reason you are telling me this?" Find out the motives involved. Then get a second opinion, check out another choice or find a different viewpoint before you plunge ahead.

Nail down all resources. It is easy to assume that formal courses and expensive, hardbound textbooks are the only way to "learn" microcomputers. In fact, these are two of the worst possible ways to become computer literate. Most of these learning aids are stillborn, hopelessly obsolete and misdirected.

Anything you can relate to that involves micros is a resource. Your first, and foremost, resource is yourself, through hands-on experience.

Other resources include micro magazines, clubs, game playing, Dungeons and Dragons sessions, micro trainers, computer stores, used wire-wrap boards, tech journals, funky books, reader-service cards, benchmarks, students, teachers, trade shows, surplus stores, computer fairs, rap sessions and swap meets.

And most important of all, go on your own vibes. There is no right or wrong direction in the micro world. In fact, 99.9 percent of the micro world remains unknown, unexplored and uncharted. So, if "they" insist on something, most often "they" don't know what they are talking about.

If you are interested in something and want to go in that direction, fine. Do it!

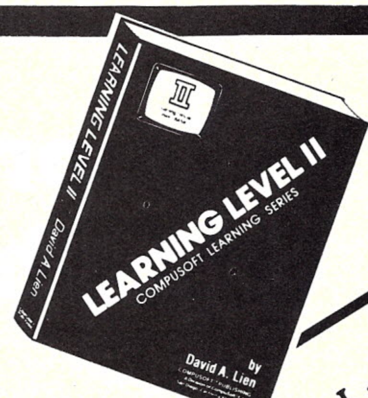
Your surest bet for long-term winning is to roll with your own vibes. Explore what you want to. Ignore the herd thundering the other way. Get off the beaten path.

Make yourself your own best customer. Satisfy your own needs and your own curiosity. Put as much psychic energy and personal value as you can in the routes that you pick, and you are certain to win the micro game.

You are, by definition, the center and the most important part of the micro universe.

Don't ever forget it. ■

HERE'S THE SECOND HALF . . .
. . . BY THE AUTHOR OF THE FIRST HALF.



360 pgs.
Soft Cover
2nd Edition

***Learning Level II* picks right up where the TRS-80* Level I manual left off, and is written in the same style that made the Level I manual a classic.**

Learning Level II teaches you to use **every Level II BASIC feature,** including **PRINT USING.** You also learn to use the built-in **Editor,** a powerful tool for changing and correcting programs. A special section covers the many changes needed to **update the Level I Manual** for use with your Level II machine.

Learning Level II also shows you how to operate the the **Interface box, Dual Cassettes, the Realtime Clock, Printers** and other peripherals. All 23 error messages are explained in detail. The entire book is written so you can understand it. (And, it has an index.)

Your Level II TRS-80* simply isn't complete without *Learning Level II.*
Order your copy today!

COMPUSOFT™ PUBLISHING
A Division of
Compusoft, Inc.
San Diego, CA 92119

Want to REALLY UNDERSTAND The BASIC Language?

From the same author comes the book you've been asking for! ***The BASIC Handbook*** is the definitive **reference** and **"idea"** book, **explaining** in detail **the BASIC language** as used in over 100 favorite micros, minis, and mainframes. A virtual **ENCYCLOPEDIA** of the BASIC language. In it is everything you need to know about the 250 most important BASIC **statements, functions, operators and commands,** explained in a way that you can put them right to work.

If there is an **alternate** way to write a **program** using other BASIC words, ***the Handbook*** shows you how. If there is a **function** needed but your machine doesn't have it, ***the Handbook*** shows you how to accomplish the same thing in other ways. **This *HANDBOOK* is written to be used!** With ***The BASIC Handbook*** you can finally **make those programs** found in magazines **run on your computer!**

360 pgs.
Soft Cover

Is TRS-80* Level II covered — YES!
Is PET covered — YES!
Is Apple covered — YES!
Sorcerer, Altair, Imsai, Etc.
YES . . . and over 50 more!

* TRS-80 is a Radio Shack Trademark.

CompuSoft Publishing—1050 E K Pioneer Way, El Cajon, CA 92020
Telephone (714) 588-0996

☐ **Yes, I want to LEARN Level II.** Please send ______ copies. My U.S. check or money order for \$15.95 each + \$1.45 for postage (air mail - \$5.00) is enclosed.

☐ **Yes, I need the BASIC Handbook.** Please send ______ copies. My U.S. check or money order for \$14.95 + \$1.35 for postage (air mail - \$5.00) is enclosed.

Name ______
Address ______
City, State, Zip ______

✓32

VISA
master charge

Dealers Inquiries Welcome

Improving the OSI Challenger C2

First in a two-part series on the inner workings of Ohio Scientific's "500" boards.

Ugo V. Ré
167 Sprucewood Drive
Levittown, NY 11756

OSI's C2 is a general-purpose system incorporating several unique features. In addition to the advertised features, there are other hardware features and prototype areas on many of the circuit boards that you can use to implement some specialized circuits.

While the documentation is sparse for the C2-4P, a thorough study of the 500 manual and the schematics for the various boards will reveal several features such as: dual system clock operation, a serial interface with multiple baud rate operation and modem control leads, two selectable video screen sizes, reverse video display and a parallel interface option.

This article describes the hardware features of the various boards and the modifications I made to these boards to implement the additional features. I used an older C2-4P system, which contained two power supplies (+5 V, 3.5 Amps and −9 V, 1.5 Amps), a four slot bus backplane, a model 500 CPU board, a model 540 video board and a model 542 polled keyboard in a typewriter-style case. In the newer systems the 500 CPU board has been replaced by a 502 CPU board, and there is only one power supply (+5 V, 4.5 Amps).

Model 500 CPU Board

This CPU board includes the following hardware features:

- A 6502A microprocessor operating at 1 MHz.
- A 6850 ACIA (asynchronous communications interface adapter)-based serial interface configured for both RS-232C and 20 mA loop current.
- 4K static RAM for user programs.
- Microsoft BASIC in 8K ROM.
- System monitor and I/O controllers in three 1702 EPROMs.
- Provisions for a user-provided 6820 PIA (peripheral interface adapter) parallel I/O port.

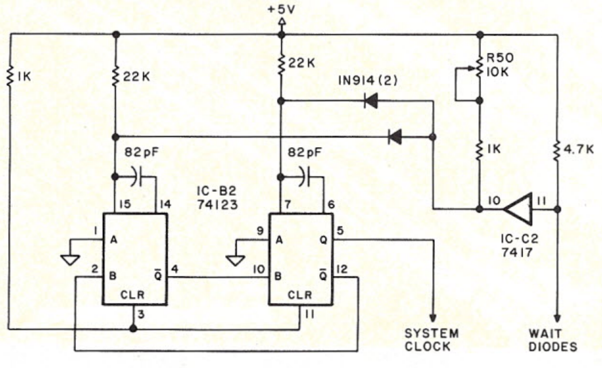


Fig. 1. CPU system clock circuit.

System Clock

The 6502 clock (see Fig. 1) is provided by a dual one-shot operating as a multivibrator. The clock circuit is populated as an adjustable two-speed clock that is normally set for a high speed of 1 MHz, but with the optional WAIT diodes, it will revert to a low-speed operation of approximately 500 kHz whenever the WAIT line is brought low.

With the components supplied by OSI, you can adjust the clock via R50 (see Fig. 2) for a high speed of 1.6 MHz without creating any problems. Since the frequency will have a tendency to change with temperature, all frequency adjustments should be made only after a long warm-up and with the case closed.

If the clock speed has been set too high or has drifted high, then the following problems, which are usually caused by the slow access time of the EPROMs and the ROMs, may occur:

- Screen does not display C/W/M? after reset.
- Monitor or I/O not operating correctly.
- Keyboard operation not recognized by computer in machine or BASIC mode.
- Programs stop running or will not run.

To correct the problem, lower the clock speed or install any of the WAIT diodes (D5, D6, D8, D10 or D11). See Fig. 3.

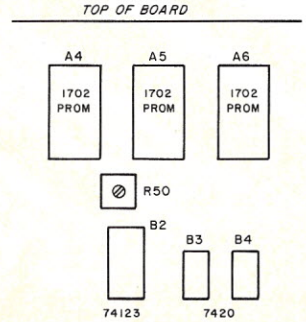


Fig. 2. System clock component location.

I have been operating the system at 1.58 MHz without any WAIT diodes and have not had a problem. However, I have noticed that programs execute faster and the cassette operation is flawless even when operated at 1200 baud.

ACIA Clock

The ACIA-based serial interface (Fig. 4) uses a 555

astable multivibrator to provide the baud rate clock. The clock circuit has provisions for five capacitors, which are jumper selected to provide a wide range of baud rates. Potentiometer R51 allows the frequency to be fine-tuned to 16 × the baud rate. However, this hard-wired method limits the interface to operation at only one baud rate.

There are two methods you can use to switch the frequency of the 555 multivibrator: change the C5 capacitor or change resistors R22 and/or R23. Of the two methods, I chose the former. Changing resistor values involves changing multiple combinations of R22 and R23 or using large values for R22. Additionally, the duty cycle of the square wave, which is directly affected by the ratio of the resistor values, becomes a spike whenever the R22–R23 ratio is too large.

I made the following modifications to provide switch-selectable baud rates:

- Calculate the capacitors required for the desired baud rates. Table 1 gives the formula to calculate the capacitor values. Table 1, column A, gives six capacitors for six possible baud rates.
- Install the six capacitors on a two-pole, six-position rotary switch (Radio Shack #275-1386). The capacitors in column A are not available as standard values. Therefore, I used standard values wired in series to obtain the required values (see Table 1, column B).
- Connect the first capacitor, .068 uF, between the first switch position and one terminal on the other half of the switch (see Fig. 5). The five remaining capacitors are wired in series from terminal to terminal starting at the first terminal.
- Remove the jumper between the J5 donut and the capacitor on the board (see Fig. 6).
- Install a shielded wire from the J5 donut to the rotary switch pole. The shield should be grounded at the board end with the other end soldered to the ground terminal on the rotary switch (see Fig. 5). I used a shielded wire to prevent stray signals from affecting the 555 circuit and frequency.
- After the wiring has been completed and checked, turn on the computer and allow it to warm up.
- To check and adjust the frequency, connect a frequency counter to L1 connector pin 7 (see Fig. 6), then reset the computer.
- Select the first switch position, 110 baud, and adjust the potentiometer, R51 (see Figs. 5 and 6), for a frequency of 1760 Hz.
- Select the other baud rates and check that they are within 0.1 percent of the required frequency (see Table 1, column C). Do not readjust R51.

Frequency Adjustment

If the frequency is higher than the required value, a low-value trimming capacitor can be added in parallel with the capacitor being tested to lower the frequency. If the frequency is lower than required, replace the capacitor with a capacitor of lower value and add trimming capacitors in parallel to obtain the required frequency.

With the capacitors wired in series, any change made to one capacitor will affect all others after it in the chain. When trimming capacitors to obtain the required frequency, always work on the lower frequency before adjusting the next higher frequency.

Using the above capacitor trimming procedure, it is possible to get the frequencies within ±.05 percent (see Table 1, column D). Generally, I use only three baud rates—110 for a Teletype and 300 and 1200 for the audio cassette and modem.

555 Frequency calculations

$$C(uF) = \frac{1}{.693 * R * fHz}$$

$$R = (R_{22} + 2(R_{23} + R_{51}))$$

(pot R_{51} set at midpoint)

$$R = 12060$$

555 Frequency components

Rate	Baud calculated value (A)	Capacitor value in series (B)	frequency calculated value (C)	frequency measured value (D)
110	.068	.068	1760	1759
300	.025	.039	4800	4801
1200	.0062	.0082	19200	19194
2400	.0031	.006	38400	38379
4800	.0015	.003	76800	76869
9600	.0008	.002	153600	153785

Table 1.

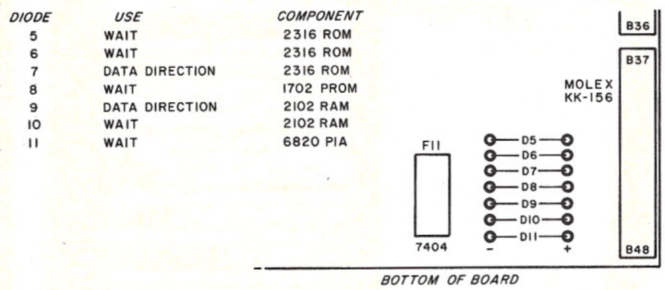


Fig. 3. WAIT diode location.

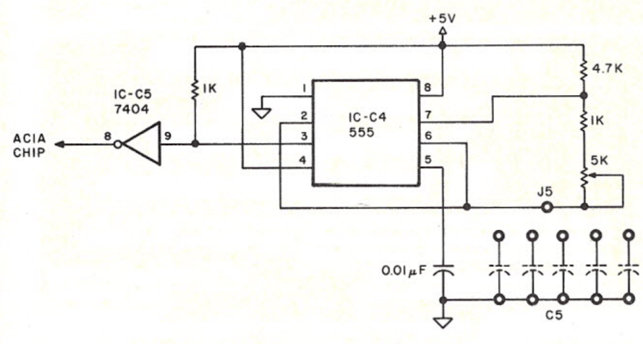


Fig. 4. ACIA clock circuit.

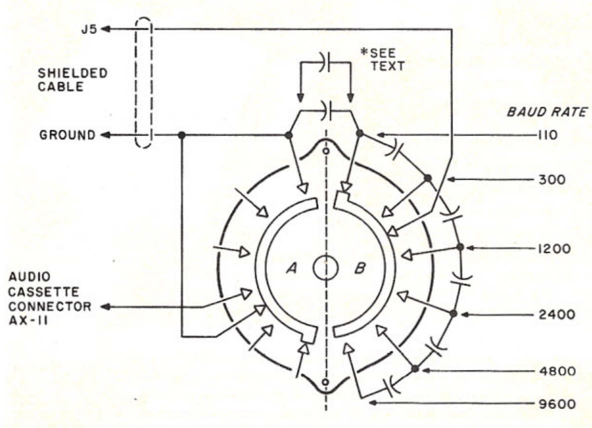


Fig. 5. Baud rate selector switch.

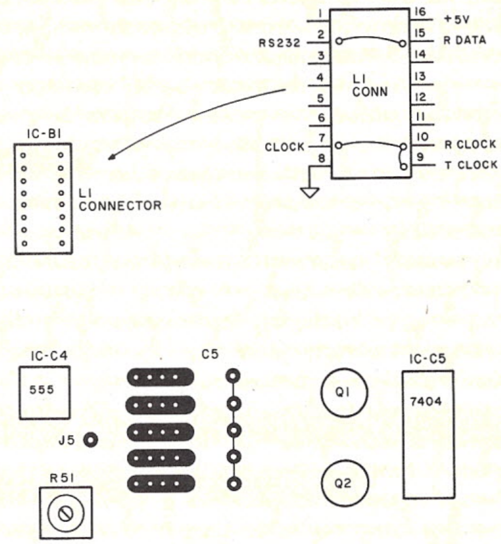


Fig. 6. ACIA clock component location.

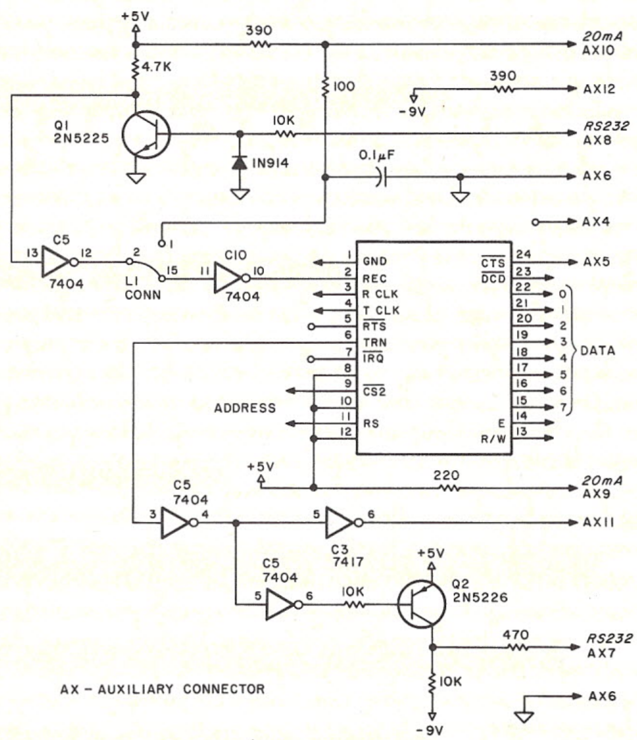


Fig. 7. Serial interface circuit.

For 300/1200 baud operation only, install a .0068 uF capacitor on the board and adjust R51 for a frequency of 19,200 Hz. Under software control, set the bits in the 6850 control register (see serial interface section) to select the ÷16 (1200 baud) or ÷64 (300 baud) clock rate.

Serial Interface

The 500 board (see Fig. 7) is provided with a 6850 ACIA chip and the components for both a 20 mA current loop and an RS-232C interface output circuit, simultaneously connected, and a 20 mA or RS-232C input circuit. Only one input circuit can be connected at a time.

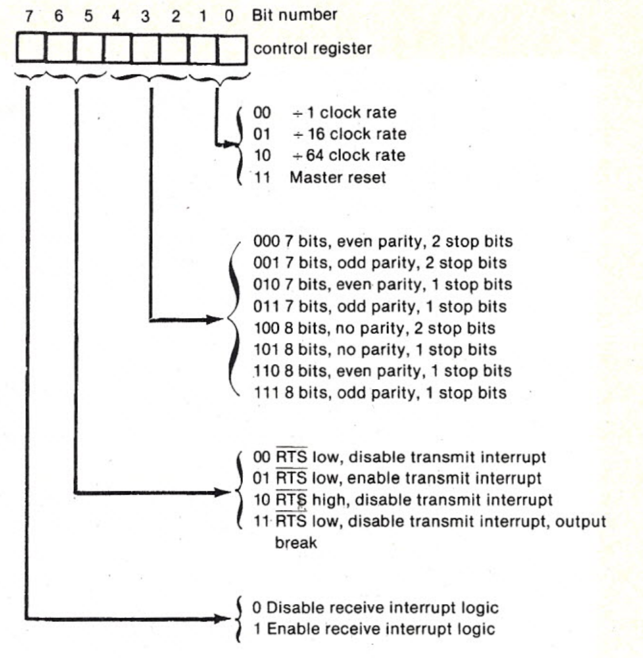
The 6850 is a 24-pin DIP that interfaces the CPU to outside devices by converting parallel I/O to serial I/O. Either a byte of parallel data or a control code is transmitted by the CPU to

the 6850, while data or status is received by the CPU. In addition to the data bus and the transmit and receive clock and data leads, the 6850 has other control leads: the modem control leads $\overline{CTS}$, $\overline{RTS}$ and $\overline{DCD}$; the RS (register select) lead, which determines which of two addressable locations will be accessed; and the R/W lead, which determines whether a read or write operation is in progress. (The two addressable locations are the control/status address (FC00 hex) and the data address (FC01 hex)).

The control register is used to control the operation of the 6850. A code, which establishes the parameters for the 6850's operation (see Table 2), is written into the control register by the CPU. When the computer is reset, the CPU first loads the register with 03 hex (reset) and then B1 hex (÷16 clock rate, eight bits, no parity, two stop bits). If this operation does not meet your needs, reset the control register and load in a new code (see Table 2).

The status register uses status flags to monitor the serial data transfer logic (see Table 3). The status register is read into the CPU's accumulator, then bit 0 or 1 is shifted right into the carry register. The carry register is then checked for the status of the receive data register or the transmit data register. If the register is set, the CPU will proceed to read or write data to the 6850 (see Table 3).

A program to read or write data to the 6850 is listed in Table 4. The other bits in the status register are used by the 6850 to determine the status of external devices, bits 2 and 3; to detect receiving errors, bits 4, 5 and 6; and to determine the source of an unacknowledged interrupt request, bit 7. The computer does not check the status of bits 2 through 7, and, in most cases, we do not need to check them for status during normal operation.



Program to change the control register.

A9; 03	LDA	Master reset code
8D; 00; FC	STA	Write code to register
A9; XX*	LDA	New parameters
8D; 00; FC	STA	Write new parameters to control register

*Note: Select new parameters from list above. Byte is then converted to hexadecimal notation and entered in program.

Example: ÷16 clock rate; 7 bits, even parity, 2 stop bits; $\overline{RTS}$ low, enable transmit interrupt; Enable receive interrupt logic 10100001 binary = A1 hex.

Table 2.

The serial I/O ports of the 6850 are wired to the components that comprise the 20 mA and RS-232C circuits. (The 20 mA current loops interface is specifically for use with the ASR-33 Teletype. It has no common ground and cannot be used with terminals requiring a common ground on output.) The 500 board has a serial interface auxiliary connector mounted near the 6850 chip (see Fig. 8), which uses pins 5 through 12 for the RS-232 and 20 mA data leads and a $\overline{CTS}$ (clear to send) control lead.

The 6850 uses a low $\overline{CTS}$ to set the status register and report a TDRE (transmit data register empty) condition. The computer determines when to transmit a byte of data to the 6850 by testing the register for the TDRE condition. A high

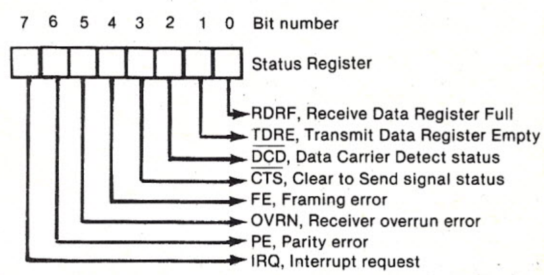


Table 3.

AD, 00, FC	LDA	load status register
4A	LSR	shift bit '0' to carry
90, FA	BCC	not ready, check again
AD, 01, FC	LDA	load receive data
9D, XX, XX	STA	store data in memory

Table 4a. Serial data operation. Receive data from the serial port. This program will continue until all data is received. It should check for end of data character, and the index register must be incremented to access the next memory address.

AD, 00, FC	LDA	load status register
4A	LSR	
4A	LSR	shift bit '1' to carry
90, F9	BCC	not ready, check again
BD, XX, XX	LDA	load Acc from memory
8D, 01, FC	STA	store data in ACIA

Table 4b. Transmit data from memory to serial port. This program will continue until all data has been transmitted. It must increment the index register to access the next memory address. It must also check for the last data address.

$\overline{\mathrm{CTS}}$ signal will prevent the register from reporting a TDRE condition.

The $\overline{\mathrm{CTS}}$ is normally strapped to ground; however, the lead can be used as an interrupt signal for loss of transmission facilities, low paper alarm, printer or cassette not on line, etc., to prevent the needless transmission of data into an open line.

To modify the $\overline{\mathrm{CTS}}$ lead, remove the strap between pin 24 and ground and install a 1k resistor from pin 5, auxiliary connector to +5 V. The $\overline{\mathrm{CTS}}$ lead, which will be high, can now be used to determine the status of the receiving device. Do not make this modification unless you can control the $\overline{\mathrm{CTS}}$ lead, since the computer will go in a loop on SAVE, where it will remain until $\overline{\mathrm{CTS}}$ is brought low.

The 6850 also has an $\overline{\mathrm{RTS}}$ (request to send) control lead used to inform a data set that it is ready to transmit data. The data set will return the $\overline{\mathrm{RTS}}$ signal as a $\overline{\mathrm{CTS}}$ signal when it establishes a data link. The 6850 outputs a low $\overline{\mathrm{RTS}}$; therefore, $\overline{\mathrm{CTS}}$ will be low and the computer will proceed to output the data. The $\overline{\mathrm{RTS}}$ terminal, pin 5 of the 6850, is not used; however, it can be wired to the spare pin 4 of the auxiliary connector.

The only remaining work is to install an EIA connector (DB255) in the cutout at the rear of the case and wire it to a Molex plug (KK-156 cut to provide one 3-pin and one 9-pin plug) following the pin-out in Table 5. Now you can use the EIA connector to connect to a printer or data set. The 20 mA leads are wired to the connector only for convenience and

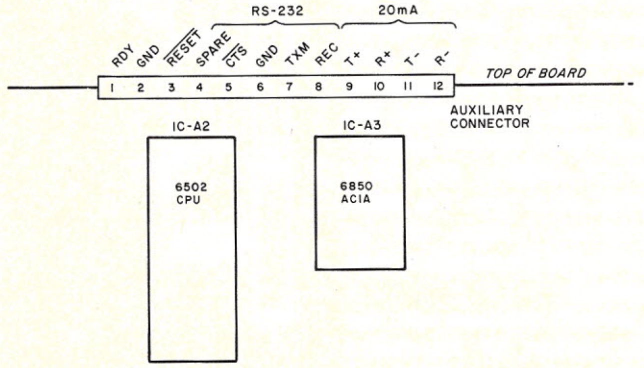


Fig. 8. Serial interface connector.

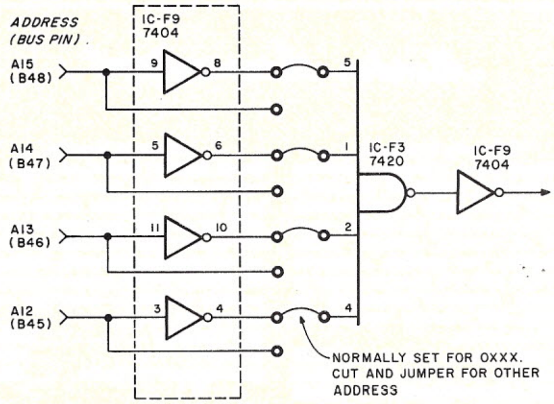


Fig. 9. 2102 RAM address decoding.

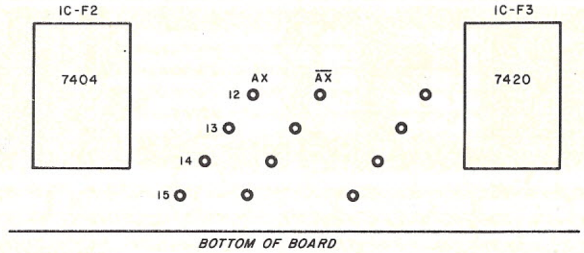


Fig. 10. Address jumpers for 2102 RAM.

EIA Pin-out		Auxiliary Connector
1	Ground	--
2	Input to CPU	8
3	Output to CPU	7
4	Request to send	4
5	Clear to send	5
7	Signal ground	6
12	20 mA input (+)	10
13	20 mA output (+)	9
24	20 mA input (−)	12
25	20 mA output (−)	11

Note: Auxiliary connector pins 1, 2 and 3 are wired to the 6502 CPU control pins, RDY and $\overline{RES}$.

Table 5. EIA connections.

are not part of the RS-232C standard.

RAM Addressing

The 2102 RAM on the 500 board is decoded for a 4K block starting at hex 0000 (page zero) (see Fig. 9). Additional memory boards, model 420 or model 527, can be added to a spare slot on the motherboard. The addition of a 420 board will not cause a problem, since the board can be decoded to occupy the next unused 4K block of memory. The addition of the 527 board, however, will create a problem, since it can only be changed to occupy an 8K block. Therefore, when installing a 527 board, you should decode it for page zero and decode the 2102 RAM for some other location. The foil trace from the $\overline{Ax}$ donut to the right-most donut (see Fig. 10) must be cut and a new jumper installed to the Ax donut.

I installed a 527 board with 16K of memory and moved the 2102 memory to location C000 hex. This memory is now used for machine-language programs. BASIC, which normally uses all consecutive memory, cannot normally access this memory; therefore, there is little danger of destroying any programs stored at that location.

ROMs and EPROMs

No changes were made to these areas, but other ROMs, PROMs or EPROMs containing a more versatile language or monitor can be substituted for the ROMs and EPROMs presently on the board. The 500 CPU manual lists other ROMs that can be used with only minor strapping changes.

Peripheral Interface

The 6820 PIA interfaces with the CPU to provide two parallel I/O ports. Although I have not implemented the parallel port fully, I have added the following components to the board (see Fig. 11):

- A 40-pin IC socket at location A1.
- J1 and J2 jumper located at the bottom and right side of socket.
- Two Molex connectors along left edge of board.

All that remains is to install the 6820 chip in the socket and then develop devices that will work with the parallel port, joystick, X-Y plotter or AC control. The newer 6522 VIA (versatile interface adapter) may be a better chip than the 6820; however, I have not investigated the modifications required.

We will continue with our discussion of hardware modifications to OSI's circuit boards next time. ■

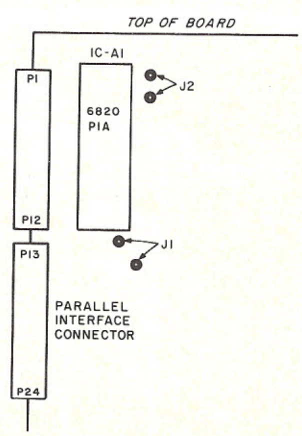


Fig. 11. PIA component location.

Back-Space Mod for CP/M and Microsoft BASIC

An annoying characteristic of the system prompted this article.

Rod Hallen
State Dept.-Accra
Washington, DC 20520

I have used Digital Research's CP/M disk operating system and Microsoft disk BASIC for about two years, and I still think the combination is great. However, one feature of both of them annoys me: the way character deletions are handled.

If you type a CP/M command line and inadvertently hit the wrong key, you can delete that character from the command line buffer by pressing the Delete key. If you continue to press the Delete key you will delete another character from the buffer for each depression. That's fine. What bothers me is the screen presentation. Instead of the cursor backing up and erasing the undesired characters, they are repeated on the screen. You type DIX by mistake, delete to get rid of the X and you have DIXX. Add the proper R and you have DIXXR. CP/M understands that this means DIR, but there has to be a better way. This is especially confusing when you have to make multiple corrections within a line.

A worse situation exists in Microsoft disk BASIC. The first time you hit Delete, a backslash is printed, then the undesired character. If you continue to hit Delete, the previous characters are printed one by one. When you hit anything other than Delete, another backslash is printed, and then you can go on with the line. I assume the reason for both methods is that they were originally designed for a hard-copy terminal, which can't back up and erase.

The Better Way

I finally decided to rewrite CP/M so that the cursor would back up and erase, instead of echo, anything I wanted to eliminate. Unfortunately, all this echoing and backslashing takes place deep within CP/M and Microsoft, and I don't have a source listing for either. I do have a source listing for my CBIOS (Custom Basic Input/Output System), which is almost as good.

I fooled CP/M by filtering the output to get rid of the echos and backslashes. At the same time, I wanted to use the Back Space key instead of the Delete key for corrections because Delete is an uppercase function on my keyboard and Back Space is not. It was simpler than I thought; now I wish I had done it a year ago.

Listing 1 is a portion of my CBIOS. Your BIOS should be similar, but it will not be exactly the same since each microcomputer implementation is different. That is why it is called a "custom" BIOS. I use a Cromemco Z-2 with an Imsai VIO video interface board and a Percom C1812 for interfacing to my Spinwriter ICSR.

CBIOS Operation

Note the CONIN routine. This keeps checking the keyboard status port to see if bit 6 is a 1. If it is, a character is waiting; if not, CONIN checks again. When a character is ready it is brought to the accumulator and bit seven is cleared. Next, check for a Back Space (hex 8) and, if you don't find one, return to CP/M with the keyboard character in the A register. If you find a Back Space, set the Delete flag for later use, then load 7FH, the ASCII Delete code, into the A register and return to CP/M.

You've hit a Back Space, the DELETE FLAG is set and the Delete code is sent because

```
CEB9 DBØ8       CONIN   IN      KSPORT  ;GET KEYBOARD STATUS
CEBB E6Ø1               ANI     4ØH     ;IS CHAR READY?
CEBD C2B9CE             JNZ     CONIN   ;IF NOT TRY AGAIN
CECØ DBØ9               IN      KDPORT  ;GET CHARACTER
CEC2 E67F               ANI     7FH     ;MASK BIT 7
CEC4 FEØ8               CPI     8       ;IS IT A BACKSPACE ?
CEC6 CØ                 RNZ             ;IF NOT RETURN
CEC7 3EØ1               MVI     A,1     ;SET
CEC9 32Ø6CF             STA     DELF    ;  DELETE FLAG
CECC 3E7F               MVI     A,7FH   ;LOAD DELETE
CECE C9                 RET
                ;
CECF C5         CONOUT  PUSH    B       ;SAVE REG
CEDØ 3AØ6CF             LDA     DELF    ;GET DELETE FLAG
CED3 FEØ1               CPI     1       ;IS IT SET ?
CED5 CAE7CE             JZ      CON1    ;IF SO, HANDLE
CED8 3AØ7CF             LDA     MICROS  ;GET MICROSOFT FLAG
CEDB FEØ1               CPI     1       ;IS IT SET?
CEDD C2FCCE             JNZ     OUTZ    ;IF NOT SKIP
CEEØ AF                 XRA     A       ;RESET
CEE1 32Ø7CF             STA     MICROS  ;   MICROSOFT FLAG
CEE4 C3Ø4CF             JMP     OUT1    ;EXIT
                ;
CEE7 79         CON1    MOV     A,C     ;GET CHARACTER
CEE8 FE5C               CPI     5CH     ;IS IT A BACKSLASH ?
CEEA C2F5CE             JNZ     CON2    ;IF NOT, SKIP
CEED 3EØ1               MVI     A,1     ;SET
CEEF 32Ø7CF             STA     MICROS  ;  MICROSOFT FLAG
CEF2 C3Ø4CF             JMP     OUT1    ;EXIT
                ;
CEF5 3EØ8       CON2    MVI     A,8     ;LOAD BACK SPACE
CEF7 CDØ3F8             CALL    ØF8Ø3H  ;AND SEND IT TO VIO
CEFA ØE7F               MVI     C,7FH   ;LOAD DELETE
                ;
CEFC 79         OUTZ    MOV     A,C     ;GET CHARACTER
CEFD CDØ3F8             CALL    ØF8Ø3H  ;SEND IT TO VIO
CFØØ AF                 XRA     A       ;RESET
CFØ1 32Ø6CF             STA     DELF    ;  DELETE FLAG
CFØ4 C1         OUT1    POP     B       ;RESTORE REG
CFØ5 C9                 RET
                ;
CFØ6 ØØ         DELF    DB      Ø       ;DELETE FLAG
CFØ7 ØØ         MICROS  DB      Ø       ;MICROSOFT BASIC FLAG
```

Listing 1. The portion of my CBIOS that handles keyboard input and screen output. This has been modified to eliminate Delete and Back Space echos and to implement a true Back Space.

CP/M doesn't know what to do with a Back Space. At this point, CP/M deletes the last character from the Command line buffer and echos it to the screen. Intercept that echo before it gets to the screen output routine, then back up and erase the last character instead.

CONOUT is my video output routine. Enter it with the desired character in the C register and immediately save it. Then get the DELETE FLAG and check to see if it is set. If neither it nor the MICROS FLAG is set, a normal character must be coming through, and you can jump directly to the screen output routine (OUTZ).

If the DELETE FLAG is set, ignore any output from CP/M and erase the last character on the current screen line. If the output character is a backslash, you're dealing with a Microsoft BASIC Delete. In this case, set the MICROS FLAG and exit without sending the backslash to the screen.

If the character isn't a backslash, it must be an echo, so ignore it. Instead, send a Back Space to the screen to back up the cursor and a Delete to eliminate the character the cursor now sits on. This is where you and I might be in conflict. Your video interface may need some other code or codes in order to back up the cursor and erase the last character. The Delete by itself might be sufficient. In any case, consult the manual for your video interface board or terminal to find out how this should be handled.

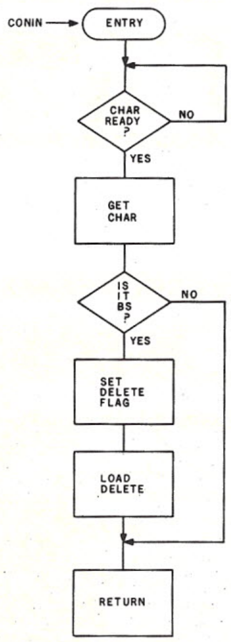


Fig. 1. Flowchart of the modified keyboard routine.

As long as you hit the Back Space key the cursor continues to back up, erasing characters as it goes. After each Delete is sent to the screen the DELETE FLAG is reset. As this routine is now written, the Delete key will still work the same way it did before, echo and all. If you want to use the Delete key instead of the Back Space key for corrections, then change the CPI 8 in line 6 of Listing 1 to CPI 7FH.

OK, but what about the second backslash in a Microsoft correction? You've already ignored the first backslash and set the MICROS FLAG. The following echoed characters will be ignored and the undesired ones erased in the same manner as with a CP/M correction. When the trailing backslash comes along, it will be ignored also because the DELETE FLAG will be reset and the MICROS FLAG will be set. At this time, reset the MICROS FLAG and exit.

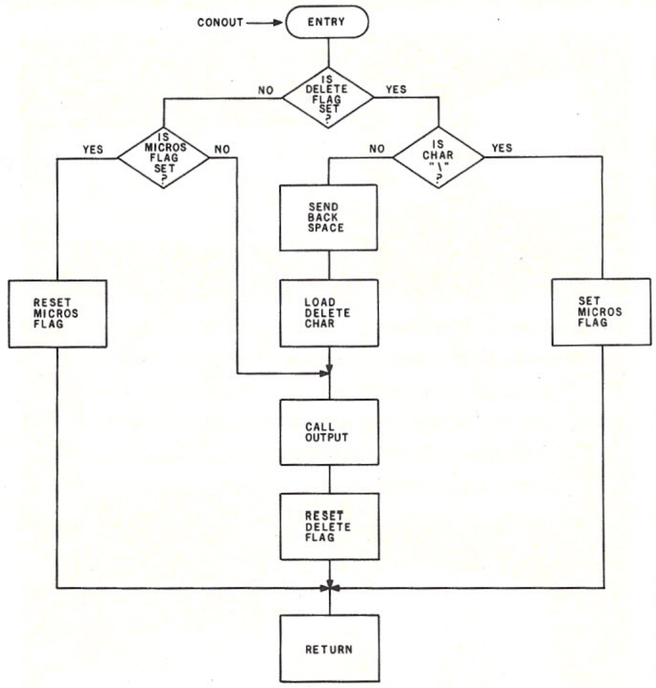


Fig. 2. Flowchart of the modified screen output routine.

An unexpected benefit of this modification came up in connection with ED.COM, the CP/M text editor. Since my new keyboard routine converts Back Spaces to Deletes, Back Space also works within the body of an ED.COM text file while editing. However, Back Space is 08H, which is also the code for a Control H, the block move code used within the body of an Electric Pencil text file.

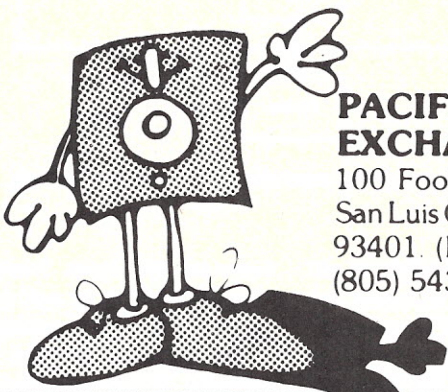
The modification I'm describing here will effectively eliminate the block move instruction, which is not good. Therefore, I have an unmodified CP/M CBIOS on the disk that contains the Electric Pencil II word processing system so it will work properly.

Figure 1 is a flowchart of the keyboard input routine, and Fig. 2 is the screen output routine.

That's all there is to it. If you have done any assembly-language programming, you shouldn't have any trouble modifying your CP/M CBIOS. I haven't mentioned the steps necessary to insert the new CBIOS into its proper place in the CP/M system and then get it onto your disks because it would take too long. ■

Dysan CORPORATION

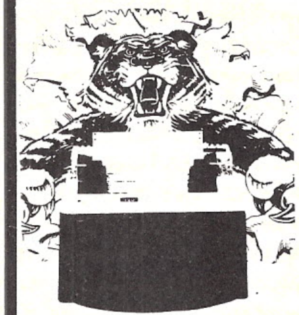
Solve your disk problems, buy 100% surface tested Dysan diskettes. All orders shipped from stock, within 24 hours. Call toll FREE (800) 235-4137 for prices and information. Visa and Master Card accepted. All orders sent postage paid.



PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA
93401. (In Cal. call
(805) 543-1037.)

✓274

SUPER SPECIAL
Apple II 16K
$999.99



The Paper Tiger.
$950.00
With Graphics $1090.00

DISK II with controller	$545.00
without controller	$465.00
MICROMODEM	$325.00
PASCAL	$425.00
ROMPLUS with keyboard filter	$149.00

16K RAMS for APPLE II TRS-80 $65

VERBATIM DISKS 10 for $27

✓105

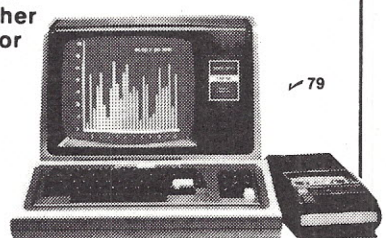
The Computer Stop
16919 Hawthorne Blvd.
Lawndale, CA 90260
(213) 371-4010

MON. - SAT.
10 - 6

TRS-80 LEVEL II
$720.
COMPLETE SYSTEM

The world's most popular microcomputer, with 16K of memory and Level 11 basic for only $720, complete with full 90 day Radio Shack warranty.
We accept check, money order or phone orders with Visa or Master Charge. (Shipping costs added to charge orders).

Disk drives, printers, peripherals, software & games... you name it, we've got it (both Radio Shack & other brands). Write or call for our complete price list.



✓79

C&S
ELECTRONICS MART, LTD.

AUTHORIZED DEALERSHIP Radio Shack®

32 EAST MAIN • MILAN, MICHIGAN 48160 • (313) 439-1508

WHY PAY MORE?
Compare our prices and service

SOROC
IQ 120.................$ 790.
IQ 140.................$1190.

CENTRONICS PRINTERS
Prices too low to advertise.
Call for best price.

LIVERMORE DATA MODEMS
300 BAUD...............$170.
RS-232, 2 yr. uncond. guarantee

BASF DISKETTES
5¼" 10/$30.
8" 10/$30.

CABLES
IEEE to Centronics.......$100.
RS-232..................$ 25.

COMMODORE SPECIALS FREE*
PET32K (N&B).....1295...175.
PET 16K (N&B)......995...135.
PET 8K.............795...100.
PET 2040 Disk
Drive..........1295...175.
PET 2022 Tractor
Printer..........795...100.
PET CZN Cassette
Deck............95... 12.

BPI GENERAL LEDGER
Accounting System for Apple II Computers—Special...$399.

CDS SERIES 80 IPS-100
- Main Frame
- Key Switch (ON/OFF)
- 8085 Processor
- 630,000 Byte Disks
- 10 Slot S-100 Buss
- 32K Ram
- 2 Independent RS-232 I/O Ports
- Up to Four Users
- 110 Volts 60 HZ
- 220 Volts Available
- List Price $3295.
- Sale Price $2700.

TO ORDER:
Send check or money order.
VISA/Master Charge/C.O.D.

*Free merchandise with purchase of CBM item.

COMPUTER PRODUCTS INTERNATIONAL

✓323

P.O. Box 17675 Washington, D.C. 20041
703-573-9633

Level II BASIC On a Z-80 System

Although the author used Radio Shack's three-ROM BASIC, the two-ROM version should work as well.

Richard J. Uschold
80 Woodview Dr.
Port Orange, FL 32019

Since I have been a dedicated hardware hacker for many years, I just had to build my own computer. I started designing at Christmas in 1976. By September 1977 I had my computer basically working, and by Christmas 1977 it was working in BASIC. It was a 2K Tiny BASIC interpreter, but it was better than nothing.

After about a year of using my Tiny BASIC, I decided I was ready for a real BASIC. Since I had chosen the Z-80 microprocessor for my computer, I could use any BASIC written for the 8080 or the Z-80.

There were a number of BASICs available that required from 8K to 24K of memory at prices from $50 to several hundred dollars. I really liked the idea of having the BASIC in ROM so that I wouldn't have to load it from tape every time, which seemed to take forever. (Even with my 2400 baud cassette interface, programs longer than 4K become annoying!) This meant I had to either use EPROMs or buy the BASIC already in ROM. The EPROMs would cost upwards of $80, plus the price of the BASIC.

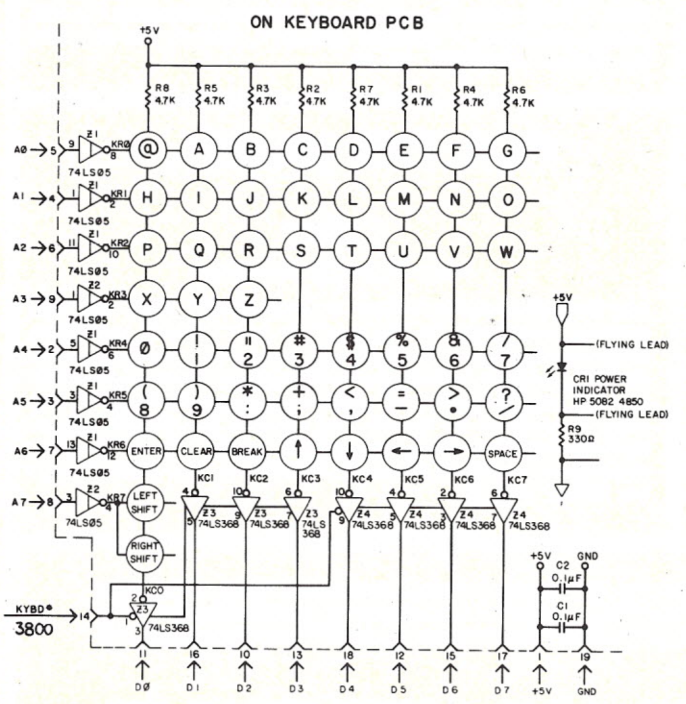


Fig. 1. TRS-80 keyboard connected to the address and data buses. (Reprinted from the "TRS-80 Technical Reference Manual," courtesy Radio Shack.)

There was only one BASIC offered in ROM that I knew of, although I had heard rumors of another one coming soon. The rumors have since become fact, and Livermore BASIC is now available on an 8K byte ROM for $95. I bought the other one, Radio Shack's Level II BASIC, for $89.10. (Several companies offer ten percent off Radio Shack's original $99 price. Radio Shack has since raised the price to $120.)

Radio Shack's Level II BASIC has another significant advantage—software availability. Since it is the most popular microcomputer around today, it has much software designed for it. Also, many programs not originally written for it are being offered in compatible forms (for example, the CP/M disk operating system and the Electric Pencil).

In this article, I will describe how I interfaced the Level II ROMs to my computer, even though my hardware bears little resemblance to that of the TRS-80. I will also give some hints to those computerists whose hardware doesn't resemble mine either!

Preliminary Work

Before I bought the Level II ROMs, I did some preliminary investigation, which included rereading articles that described the TRS-80 hardware and software. I also bought and read the "TRS-80 Microcomputer Technical Reference Handbook" published by Radio Shack. All of this material provided several important pieces of information.

First, the TVT was a more or less standard type of memory-mapped interface, which, I figured, should present no problems.

Second, the keyboard was an unorthodox arrangement with the key matrix directly mapped in memory (see Fig. 1). I figured I could write a program to take ASCII data from my keyboard and calculate the required memory bits to set so that the ROM could find the bits in memory and convert them back to ASCII (a kludge, but it worked!).

Third, the cassette interface was software timed and would require a different clock rate on my processor or else some software patches to get the timing right.

Finally, and perhaps most importantly, the ROMs were located in memory at address 0000H. This meant I would have to move my monitor, which was now there, to another address. I moved it to F000H. This required a reset vector other than 0000H to initialize to the monitor.

The circuit I used was described in the September 1977 *Kilobaud* ("Using an Invisible PROM," p. 106, by Jack Regula). My version is in Fig. 2. I spent the next month or so rewriting and improving my monitor. When I had it just right, I put it in

```
;TRS-80 KEYBOARD SIMULATOR INTERRUPT ROUTINE
;BY RICHARD J. USCHOLD
        ORG 0F776H
TKBINT  PUSH HL         ;SAVE REGISTERS
        PUSH AF
        PUSH BC
        LD A,3CH        ;MASK KBRD AND LOWER INTS
        OUT (INTMSK),A  ;FOR MY SYSTEM ONLY
        EI
        XOR A           ;CLEAR A
        LD B,8          ;LOOP COUNT
        LD HL,3801H     ;KBRD MEMORY ADDRESS
CLRLOP  LD (HL),A       ;CLEAR KBRD ADDRESS
        RLC L           ;GENERATE NEXT ADDRESS
        DJNZ CLRLOP
        LD L,0FFH       ;KEY PRESSED BYTE
        LD (HL),A       ;CLEAR IT
        IN A,(KBRD)     ;GET DATA FROM KEYBOARD
        LD C,A          ;SAVE DATA
        BIT 7,A         ;CHECK ALL SHIFT BIT
        JR Z,NASHFT     ;NOT ALL SHIFT
        LD A,1          ;YES-SHIFT BIT DATA
        LD L,80H        ;SHIFT ADDRESS
        LD (HL),A       ;SET SHIFT BIT
        RES 7,C         ;CLEAR ALL SHIFT BIT
        LD A,C          ;GET DATA
NASHFT  CP LFBRKT       ;UPPER LIMIT CHARACTER
        JR NC,VALIDQ    ;ONLY ONE VALID ABOVE THIS
        CP ATSN         ;CHECK IF ALPHABETIC
        JR C,NOALPH
        RRCA            ;/2 - YES,GENERATE
        RRCA            ;/4 - ADDRESS BIT
        RRCA            ;/8
        AND 03          ;MASK ALL BUT TWO BITS
        INC A           ;ADJUST COUNT TRUE
        LD B,A          ;SET UP LOOP COUNT
        XOR A           ;CLEAR A
        SCF             ;CARRY TO BE SHIFTED IN
GENADR  RLA             ;GENERATE ADDRESS BIT
        DJNZ GENADR
        LD L,A          ;SAVE ADDRESS
DATA    LD A,C          ;RESTORE ASCII DATA
        AND 07          ;MASK ALL BUT THREE BITS
        LD B,A          ;MOVE TO COUNTER
DATA1   XOR A           ;CLEAR A
        SCF             ;CARRY TO BE SHIFTED IN
        INC B           ;MAKE COUNT TRUE
GENDAT  RLA             ;GENERATE DATA BIT
        DJNZ GENDAT
        LD (HL),A       ;SET BIT IN MEMORY
        LD L,0FFH       ;KEY PRESSED BYTE
        LD (HL),A       ;SET KEY PRESSED BIT
NVALID  POP BC          ;RESTORE REGISTER
        JP KBINT1       ;FINISH BY DOING NORMAL
;KEYBOARD INTERUPT ROUTINE
NOALPH  CP '!'          ;IS IT CONTROL?
        JR C,CNTRL      ;YES
        BIT 3,A         ;NUMERIC OR SPECIAL?
        JR NZ,NONUM
        BIT 4,A         ;CHECK IF SHIFT
        LD B,10H        ;SAVE ADDRESS BIT
        JR NZ,NOSHFT
SHFT    LD L,80H        ;SHIFT ADDERSS
        LD A,1          ;SHIFT BIT
        LD (HL),A       ;SET SHIFT BIT
NOSHFT  LD L,B          ;SET ADDRESS BIT
        JR DATA
NONUM   LD B,20H        ;SAVE ADDRESS BIT
        AND 14H         ;CHECK IF SHIFT
        JR Z,SHFT
        XOR 14H         ;CHECK IF SHIFT
        JR Z,SHFT
        JR NOSHFT
VALIDQ  CP RUBOUT       ;THIS IS BACK ARROW KEY
        JR NZ,NVALID
CNTRL   LD HL,CTRLTB    ;TABLE ADDRESS
        LD BC,8         ;LOOP COUNT
        CPIR            ;SEARCH TABLE FOR MATCH
        JR NZ,NVALID    ;NOT FOUND
        LD HL,3840H     ;CONTROL BIT ADDRESS
        LD B,C          ;LOAD LOOP COUNT
        JR DATA1        ;COMPUTE BIT AND FINISH
CTRLTB  DEFB SPACE
        DEFB RTAROW
        DEFB RUBOUT
        DEFB LF
        DEFB UPAROW
        DEFB ESC
        DEFB ENQ
        DEFB CR
INTMSK  EQU 18H
KBRD    EQU 4
LFBRKT  EQU 5BH
RUBOUT  EQU 7FH
SPACE   EQU 20H
RTAROW  EQU 9
LF      EQU 10
UPAROW  EQU 0BH
ESC     EQU 1BH
ENQ     EQU 5
CR      EQU 0DH
ATSN    EQU 40H
;-----------------------
;THIS IS THE NORMAL KEYBOARD INTERUPT SERVICE ROUTINE
        ORG 0FC9CH
KBDINT  PUSH HL
        PUSH AF
KBINT1  LD HL,KBRDAT    ;SAVE ADDRESS FOR DATA
        IN A,(KBRD)     ;GET DATA
        LD (HL),A       ;SAVE IT
        CP CTRLZ        ;TO RETURN TO MONITOR
        LD A,10H        ;ENABLE ALL INTERUPTS
        OUT (INTMSK),A
        JR Z,TOMON      ;IT WAS CONTROL Z
        DEC HL          ;POINT TO STATUS WORD
        SET 0,(HL)      ;SET KBRD FLAG
        POP AF
        POP HL
        EI              ;ENABLE INTERUPTS
        RET             ;RETURN FROM INTERUPT
TOMON   POP AF          ;RESTORE REGISTERS
        POP HL          ; FOR SAVE ROUTINE
        JP RGSAVE       ;SAVE REGISTERS AND GO TO MONITOR
KBRDAT  EQU 0F03AH
CTRLZ   EQU 1AH
RGSAVE  EQU 0F8C9H
        END
```

Listing 1. TRS-80 Keyboard Simulator program converts the ASCII data from my keyboard to the memory-mapped bits expected by the Level II BASIC ROM. Program is simpler than it might have been due to the logical placement of the keys in the keyboard matrix (see Fig. 1).

EPROM, and I ordered the Level II ROMs.

Getting Ready

While waiting for the ROMs to arrive, I wrote a couple of programs to simulate the TRS-80 hardware, and I made a couple of hardware modifications to my computer in those areas that could not be readily done with software. The first program, in Listing 1, simulated the TRS-80 memory-mapped keyboard. This program is an interrupt driver that must be used as such. The program exits by jumping to my normal keyboard interrupt routine.

As you can see, the normal routine checks for a control-Z character and jumps to the monitor if it detects one. This is an invaluable feature of my monitor. This allows me to always jump back to the monitor if for some reason the executing program hangs up (except if it disables interrupts or destroys the monitor RAM area).

If you don't have an interrupt-driven keyboard, you can't use the program in Listing 1, but don't worry, you can still put Level II on your computer. It is highly desirable that you have some method of interrupting the computer, saving the registers, etc., and jumping back to your monitor. It is also necessary that you use interrupt mode 2 on the Z-80, since the other interrupt locations are used by Level II BASIC.

If you use Listing 1 with most keyboards, you will not be able to enter the same character twice in a row! The reason for this is because when the program sets the bits in memory to simulate the TRS-80 keyboard, it never resets the bits until the

```
00100 ;BASIC INITILIZATION ROUTINES
00110 ;RICHARD J. USCHOLD
00120 PCG     EQU 1CH
00130 SPACE   EQU 0F82FH
00140 CHIN    EQU 0F859H
00150 JPIGRT  EQU 0F5CFH
00160 VIDVEC  EQU 401EH
00170 INITVT  EQU 0FD00H
00180 TVT     EQU 14H
00190 PRTVEC  EQU 4026H
00200 LINEPP  EQU 4028H
00210 KBDVEC  EQU 4016H
00220 JMP4    EQU 0FC59H
00230 KBDST   EQU 0F039H
00240 RUBOUT  EQU 7FH
00250 BS      EQU 8
00260 ENQ     EQU 5
00270 CHRCNT  EQU 402AH
00280 TYPOUT  EQU 0F86BH
00290 OK      EQU 0F84AH
00300 L2VID   EQU 33H
00310 TRHDLR  EQU 4012H
00320 INTMSK  EQU 18H
00330 BOOT    EQU 69FH
00340 SVPC    EQU 0F051H
00350 DELAY   EQU 60H
00360 CRET    EQU 0DH
00370 TRSCAS  EQU 0FFH
00380 VIDJMP  EQU 0F06BH
00390 UPAROW  EQU 5BH
00400 NMIVEC  EQU 66H
00410 ;GENERATE TRS-80 GRAPHICS
00420         ORG 0F5E9H
00430 BASIC   IN A,(PCG+2)  ;DISABLE WRITE PROTECT ON
00440 ;PROGRAMMABLE CHARACTER GENERATOR
00450         LD HL,33FFH   ;LAST PRG CHR ADDRESS
00460         LD C,0FFH     ;DATA FOR LAST CHARACTER
00470         LD D,40H      ;64 CHARACTER COUNT
00480 MAINLP  LD E,4        ;4 DOT ROWS PER CHAR COUNT
00490 CHARLP  LD B,2        ;SHIFT LOOP COUNT
00500 SHFTLP  RLC C         ;GET DATA TO CARRY
00510         RRA           ;ROTATE CARRY TO ACC
00520         SRA A         ;COPY BIT TO FOUR PLACES
00530         SRA A
00540         SRA A
00550         DJNZ SHFTLP   ;DO NEXT FOUR BITS
00560         LD B,4        ;4 LINES PER DOT ROW COUNT
00570 DOTLOP  LD (HL),A     ;LOAD DATA TO PRG CHR
00580         DEC HL        ;BUMP TO NEXT ADDRESS
00590         DJNZ DOTLOP   ;DO 4 LINES
00600         DEC E         ;CHARLOOP COUNTER
00610         JR NZ,CHARLP
00620         LD A,C        ;GET DATA FOR NEXT ROW
00630         SUB 41H       ;GENERATE NEXT DOT ROW
00640         LD C,A        ;SAVE NEXT DOT ROW
00650         DEC D         ;MAIN LOOP COUNTER
00660         JR NZ,MAINLP
00670         IN A,(PCG+3)  ;PROTECT MEMORY
00680 ;BASIC COMMAND DECODE - THE NEXT TWO LINES ARE PARTICULAR TO MY MONITOR
00690         CALL SPACE    ;TYPES A SPACE
00700         CALL CHIN     ;GET A CHARACTER FROM KEYBOARD AND ECHO IT
00710         CP 'C'        ;FOR CONTINUE
00720         JR Z,RETBAS ;GO BACK TO BASIC
00730         CP 'I'        ;FOR INITIALIZE
00740         JP Z,0001     ;INITIALIZE BASIC
00750         CP 'R'        ;FOR RESET
00760         JR NZ,JPIGRT  ;ILLEGAL CHARACTER
00770 ;THIS ROUTINE PRINTS ILLEGAL CHARACTER MESSAGE AND RETURNS TO THE MONITOR FOR THE NEXT COMMAND
00780         JP NMIVEC     ;TRS-80 RESET SWITCH
00790         NOP
00800         NOP
00810         NOP
00820         NOP
00830 RETBAS  LD HL,VIDPCH ;INIT VIDIO PATCH
00840         LD (VIDVEC),HL ;CHANGE TRS VECTOR
00850         CALL INITVT   ;THIS SETS UP MY TVT AND
00860 ;CLEARS THE SCREEN. THIS IS PARTICULAR TO MY TVT AS IS THE NEXT LINE
00870         LD A,0CEH     ;NO SCROLL, CURSER OFF
00880         OUT (TVT+2),A
00890 ;THIS NEXT SECTION SETS UP A JUMP ADDRESS SO I CAN SWITCH BETWEEN THE NORMAL SPACE
00900 ;COMPRESSION CODES OR 64 MORE PROGRAMMABLE CHARACTERS
00910         LD HL,VIDJMP  ;JUMP TO 04A6 DOES SPACE
00920         LD (HL),0C3H  ;COMPRESSION CODES. JUMP
00930         INC HL        ;TO 047D DOES PROGRAMMABLE
00940         LD (HL),0A6H  ;CHARACTERS. FROM BASIC,
00950         INC HL        ;POKE -3988,125 FOR PRG
00960         LD (HL),04H   ;CHRS, POKE -3988,166
00970                       ;FOR TABS
00980         LD HL,TRSPRT  ;PRINTER DRIVER
00990         LD (PRTVEC),HL
01000         LD HL,KBDSUB    ;KEYBOARD SUBTITUTE DRIVER
01010         LD (KBDVEC),HL  ;CHANGE TRS VECTOR
01020         LD A,57
01030         LD (LINEPP),A ;PRINTER LINES PER PAGE
01040         NOP
01050         JP JMP4            ;THIS IS THE SECTION IN
01060 ;MY MONITOR WHICH RESTORES THE REGISTERS AND RETURNS TO THE MAIN
01070 ;PROGRAM - AS FROM A CONTROL Z INTERUPT.
```

Listing 2. The first part of this program generates the bit patterns necessary to program my programmable character generator so it simulates the TRS-80 graphics. The second part sets up my computer so it is compatible with the Level II BASIC ROM.

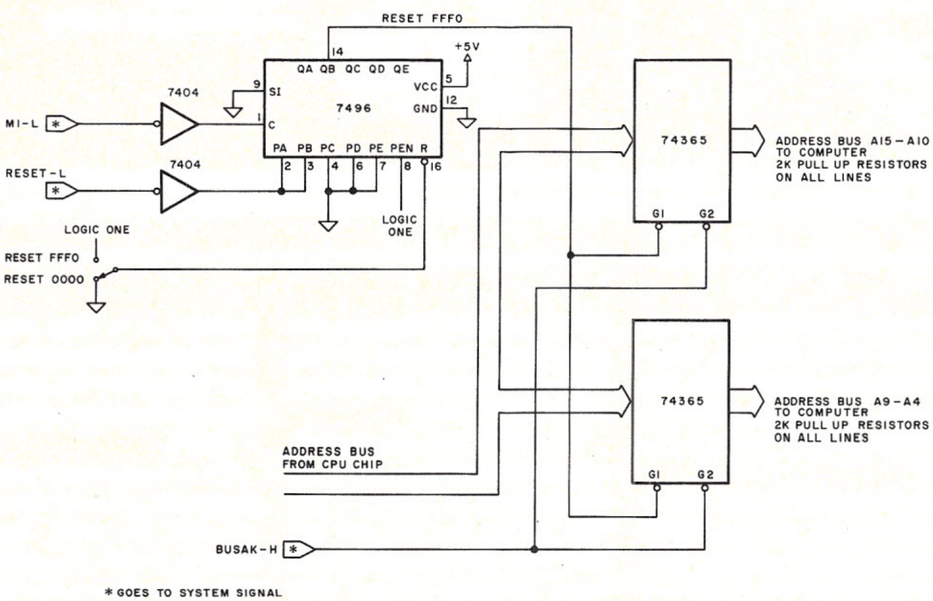


Fig. 2. Alternate reset vector circuit. Address FFF0 must be in PROM and contain a three-byte jump instruction to the start of the monitor.

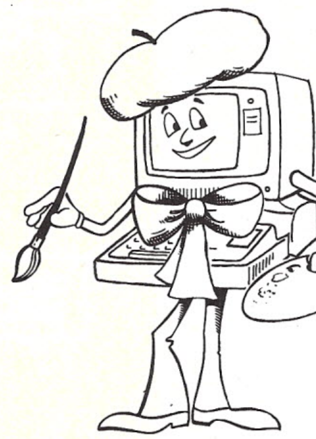
next key is hit. If the next key is the same as the last one, the same bits will be set and the ROM will think you have not released the key yet!

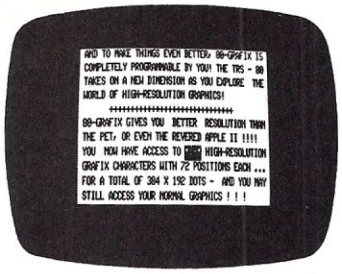
There are several solutions to this problem. I modified my keyboard so it gives a second data strobe when a key is released. This will strobe in a null, and the program will clear the memory when the key is released. Another solution is to hit any key on the keyboard that is not encoded by the program. This will clear the memory and leave it that way. This is only necessary if you wish to hit the same character twice in a row.

Actually, I don't really recommend you use this program. I am only describing it since it is the way I started this project. Later,

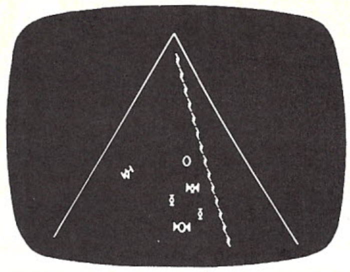
FROM PROGRAMMA
HI-RESOLUTION GRAPHICS FOR THE TRS-80®

LOWER CASE
The 80-GRAFIX board includes two sets of lower case characters at no additional cost.





INVERSE VIDEO
The 80-GRAFIX board allows you to do inverse video to high-light your screen displays.



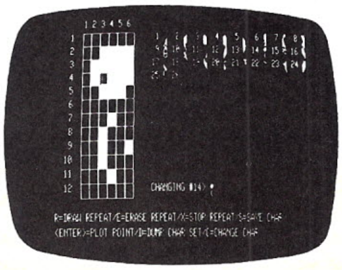
DEMONSTRATION PROGRAMS
The 80-GRAFIX board is supplied with a Character Generator software and several demonstration programs.

FINALLY, AT LAST. . .

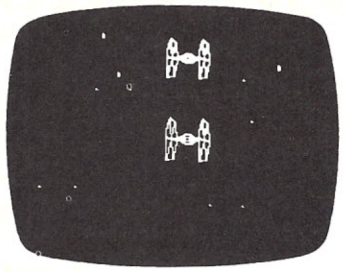
HI- RESOLUTION GRAPHICS is available for your TRS-80 computer system. The 80-GRAFIX board from PROGRAMMA International, Inc. gives your TRS-80 high resolution capability that is greater than the Commodore CBM/PET or even the revered APPLE II.

80-GRAFIX gives the TRS-80 an effective screen of 384X192 pixels, versus the normal 127X192 for the TRS-80, 80X50 for the CBM/PET, or the 280X192 of an APPLE II. As an added feature, 80-GRAFIX offers you lower case characters at no additional cost. Of course, you can also create your own set of up to 64 original characters using the supplied Character Generator software.

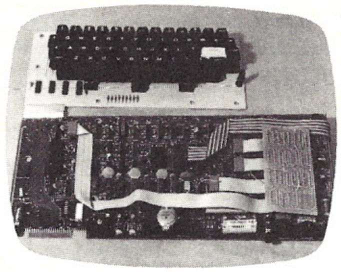
The 80-GRAFIX board is simple to install (note that this voids your Radio Shack warranty), and programming is done through BASIC. 80-GRAFIX opens up a whole new realm of software development and excitement never dreamed of for the TRS-80!



CHARACTER GENERATOR
The supplied character generator software allows you to create your own character set of up to 64 original characters.

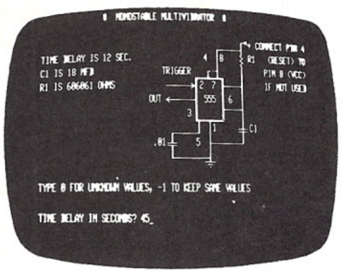


REAL-TIME GRAPHIC GAMES
With the 80-GRAFIX board you can write exciting real-time games using BASIC.

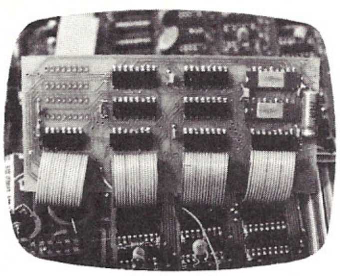


EASY INSTALLATION
The 80-GRAFIX board is simple to install and fits inside the TRS-80 case.

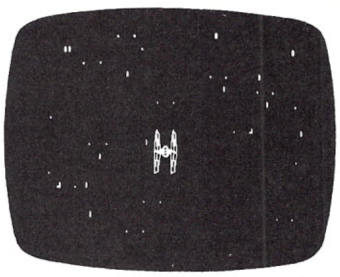
GRAPHICS GALORE
The 80-GRAFIX board and the supplied Character Generator allow you to become an artist.



ELECTRONIC DESIGN
The 80-GRAFIX board has unlimited application in Electronic design and Education.



80-GRAFIX HI-RESOLUTION
Finally, the only means to protect your computer investment is to order an 80-GRAFIX board TODAY!



EXCITEMENT & FUN
Open up a new realm of software development with the 80-GRAFIX board.

Available exclusively through PROGRAMMA at the cost of $149.95
Please check with us for availability prior to ordering
VISA and MASTERCHARGE accepted
TRS-80 is a registered trademark of the Tandy Corp.

PROGRAMMA INTERNATIONAL, INC.
3400 Wilshire Blvd.
Los Angeles, CA 90010 ✓11
(213) 384-0579 • 384-1116 • 384-1117

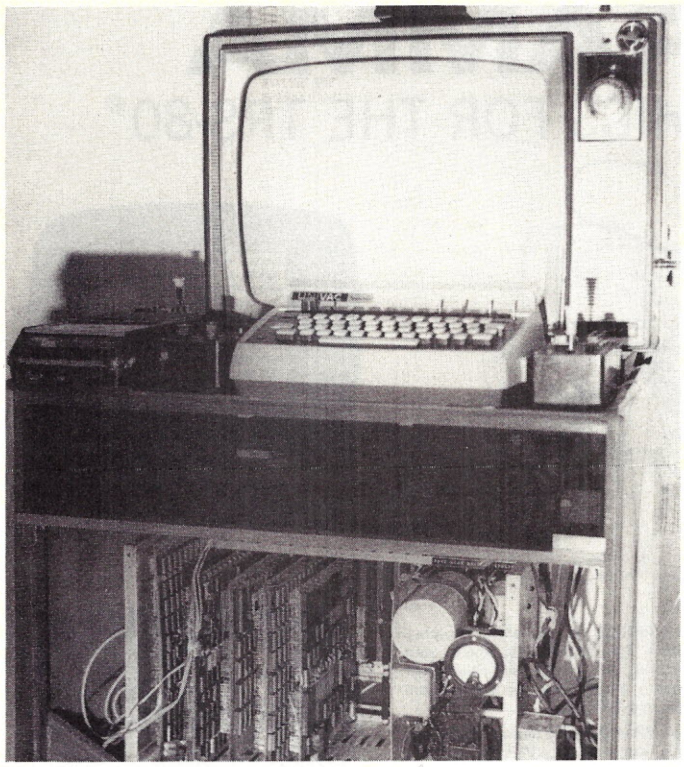


Photo 2. *My completely home-brew system. The first board contains the TVT and programmable character generator. Board 2 has my front panel logic, the interrupt logic, EPROM programmer, two serial ports and the cassette interface, which supports Kansas City Standard, Tarbell, PE2400 Radio Shack Level II and CUTS with a slight mod. The third board contains the Z-80 CPU chip, 10K of static RAM, 3K EPROM, the clock switch and "No Memory" interrupt circuit. Board 4 is a 12K static RAM board. The fifth board contains a joystick interface, Level II ROMs, alternate reset circuit, floppy disk interface, real-time clock and sockets for 32K of dynamic RAM.*

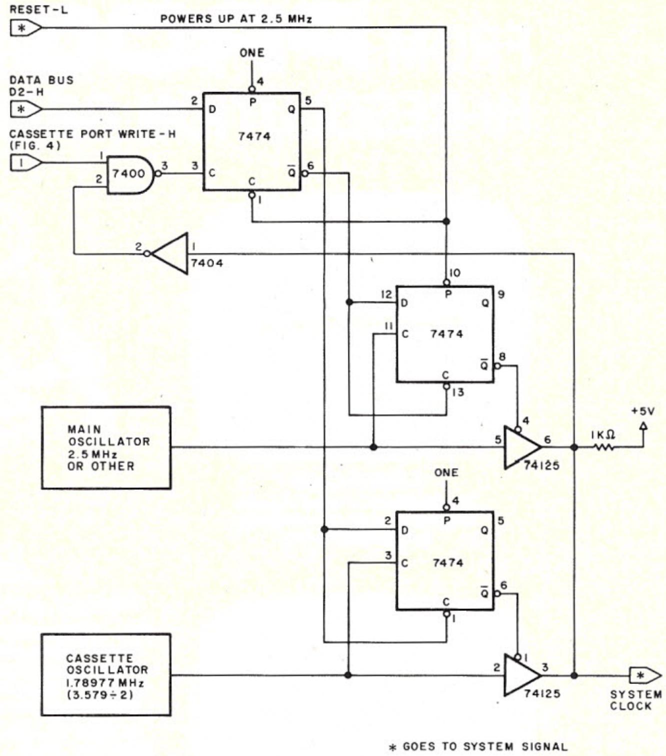


Fig. 3. Clock switch circuit automatically switches the clock from the normal frequency (2.5 MHz on my system) to 1.79 MHz when I/O port FF is written with bit 2 set. It switches back when port FF bit 2 is reset. This bit is the TRS-80 cassette motor control bit.

I'll tell you what you should use and what I am now using.

Another noteworthy feature about this program is the shift. The TRS-80 keyboard program generates lowercase characters if the shift key is pushed with a regular key. It also generates special control characters when the shift is pushed with the arrow keys.

I handled this by using the eighth bit as the shift bit. My keyboard has an extra key that sets the eighth bit when pushed. Most keyboards don't have this.

The second program I wrote while waiting for the ROMs is an initialization of my system so that the ROMs will think they are hooked up to a TRS-80. Listing 2 essentially is the program, although it is a little bit different. I changed it slightly after I got the ROMs and learned a few things I didn't originally know.

The first part of the program initializes my programmable character generator to simulate the TRS-80 graphics characters. The programmable character generator is essentially the same as the one described in *Byte* magazine (May and June 1978). There are 128 programmable characters that can be printed by sending the codes 80H–FFH to the video driver or directly loading these codes in

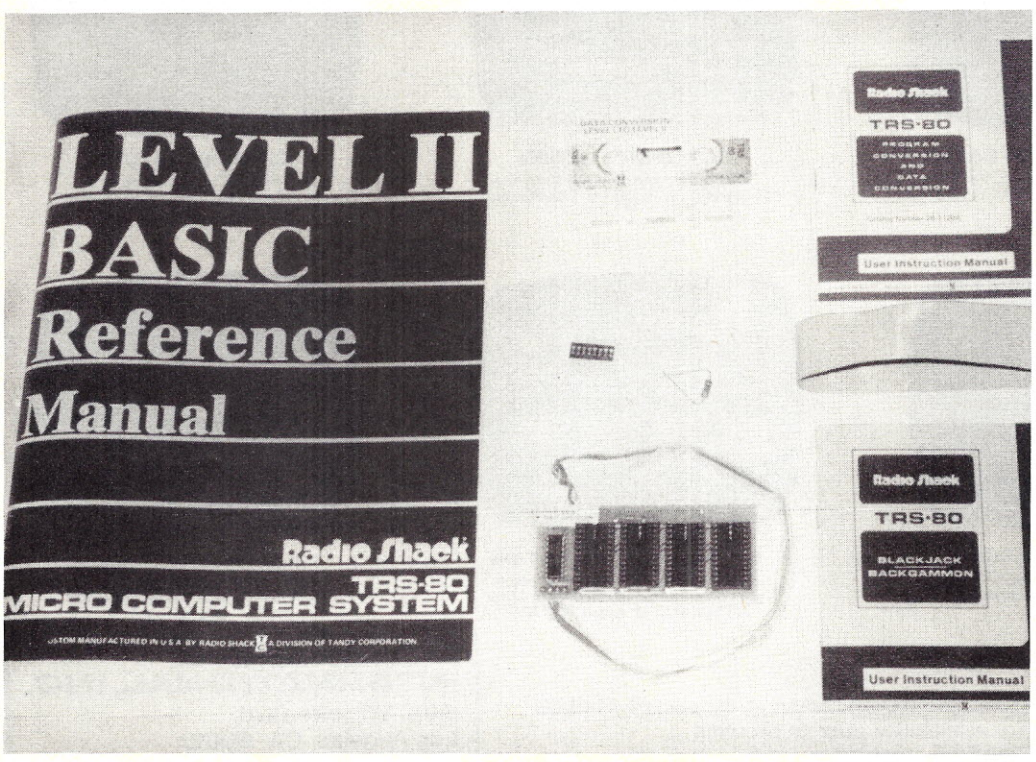


Photo 1. Level II kit. ROMs have been removed from the circuit board. (Photos by Michael Tabellion).

the TVT memory area. The TRS-80 has 64 graphics characters having codes 80H–BFH. These corresponding characters are generated by the program.

The next section, command decode, checks for one of three options: initialize, continue or reset. The initialize section jumps to the ROM so it can initialize the Level II RAM area as it requires. The reset jumps to the ROM, where the reset button on the TRS-80 would send it. This is used when the Level II hangs up and you do not wish to destroy the BASIC program in memory.

On my system, I type a control-Z to get back to the monitor and then BR. B is the BASIC command in my monitor that jumps to the program I am now describing. R is the reset option. The continue option initializes a few more things, which I'll describe later, restores the registers and continues where it was interrupted (usually by a control-Z). I frequently use this to save BASIC programs with my 2400 baud cassette interface rather than use Level II's 500 baud cassette interface.

I made several hardware mods to accommodate the Level II ROMs. The simplest was to move my RAM, EPROM and TVT RAM to the proper locations. The TRS-80 hardware manual has a memory map, so this was no real problem. The other two mods were a bit more involved. Both of these mods are for the cassette interface.

The first one (Fig. 3) changes the clock speed during the cassette operation. Normally my computer runs at its rated speed of 2.5 MHz; during a cassette operation, the speed is reduced to 1.7898 MHz. This is about one percent higher than the TRS-80 clock and is more than close enough when you consider the tolerance of the cassette machine.

The required clock rate is one-eighth the rate of my TVT clock, so I didn't require another oscillator. The required clock is also one-half the color burst frequency. There are inexpensive crystals available that you can use; 3.579 MHz color burst crystals cost less than $2.

The other changes are more directly related to the cassette interface itself (Fig. 4). The output circuit is little more than a couple of latches and a few resistors. I also added some Tri-state buffers so I could use the same cable as my 2400 baud interface. The first input circuit I tried is simpler than what the TRS-80 has, with three fewer op amps and many fewer resistors and capacitors. The idea was to change the input circuit I had been using with my 2400 baud interface as little as possible.

Well, I was finally ready for the ROMs, which would not arrive for over a month.

The ROMs Arrive

After calling the company twice, asking where my order was, I finally received the ROMs, which came on a small circuit board with a 24-pin jumper cable and another four-wire ribbon cable. No instructions came with the kit; however, the handbook shows a schematic of the circuit board (Fig. 5). There are also other items, including an unprogrammed DIP header and a resistor, in the kit (see Photo 1). The DIP header alters the ROM decode in the TRS-80; I'm not sure what the resistor is used for. Anyway, I didn't use either of these.

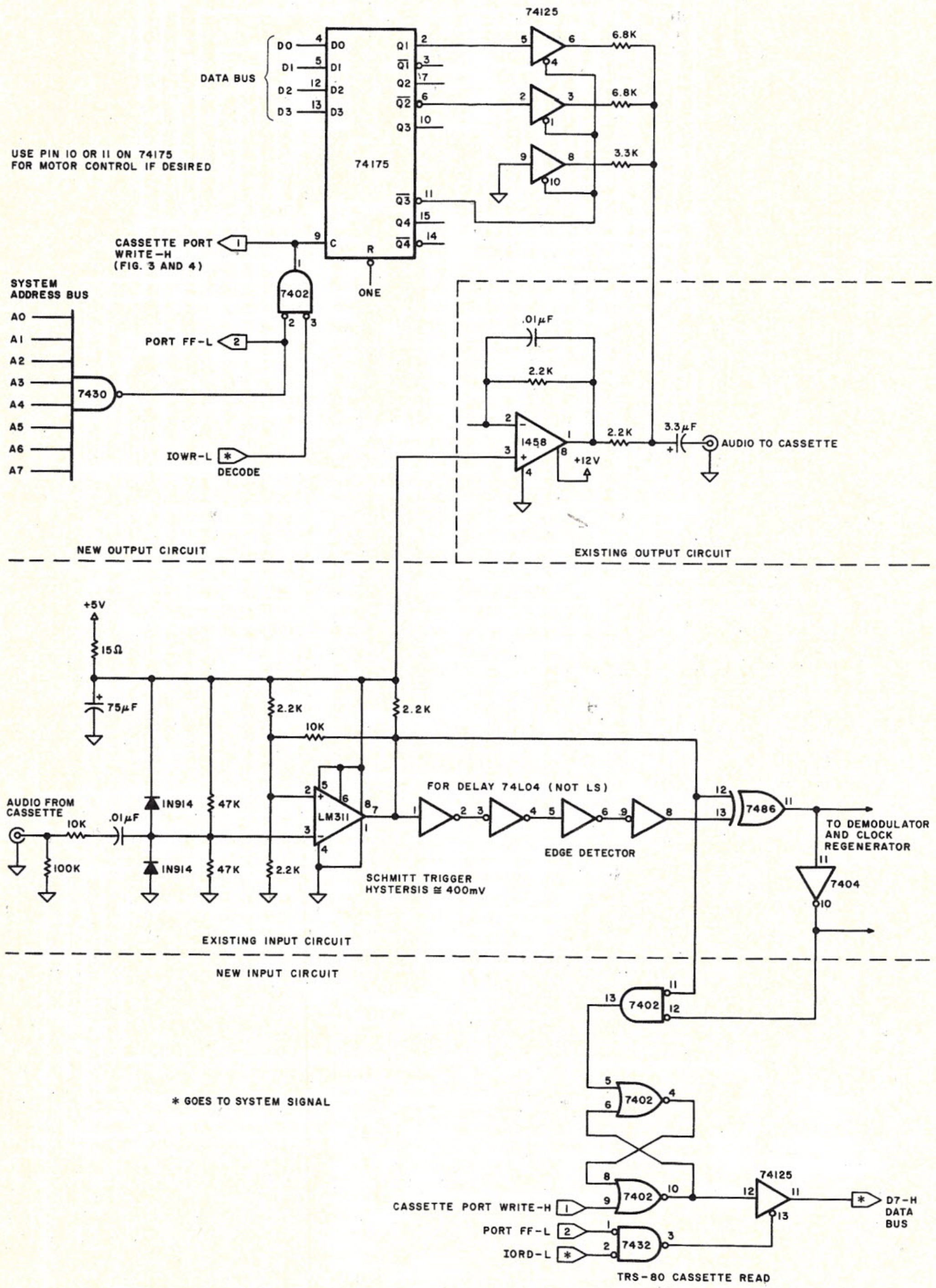


Fig. 4. Cassette output circuit similar to the TRS-80. I added the Tri-state buffers and changed the resistor values a bit so I could wire it directly to my existing output circuit. You can use bit 2 for cassette motor control if you wish.

Also included are three prerecorded cassettes with some very brief instructions on how to use them. One cassette contains Blackjack and Backgammon. The other two cassettes are for conversion of Level I programs and data to Level II format. I haven't had a need for these two yet, though I have used the games a few times. Finally, there is the "Level II Reference Manual," along with errata sheets, containing useful information.

The small circuit board didn't seem to fit anywhere in my system, so I wired up three sockets and just removed the ROMs. A friend had given me a poor copy of a copy containing a hex dump of the ROMs and partial disassembly of the initialization portion of the program. The first thing I did was to check the first few bytes in each ROM. They matched! Next, I ran off a hex dump of my own so I could read it without straining my eyes.

There was one more thing I wanted to do before I actually tried to execute the program contained in the ROMs. From all the information I had acquired, I knew that the TRS-80 used interrupts only when it had the expansion interface connected. Also, it only used interrupt mode 1 on the Z-80 chip. Since my system would only work if I used interrupt mode 2, I searched the ROMs for any instructions that affected the interrupts. There were two: a disable interrupts at 0000H and an enable interrupts at 06E4H.

The enable interrupt instruction is actually the interrupt service routine, which is moved to RAM during the initialization. The routine merely enables interrupts and returns. This is modified when interrupts are needed. What all this boils down to is that I shouldn't have any problems with my interrupt-driven keyboard as long as I start the ROM at 0001H.

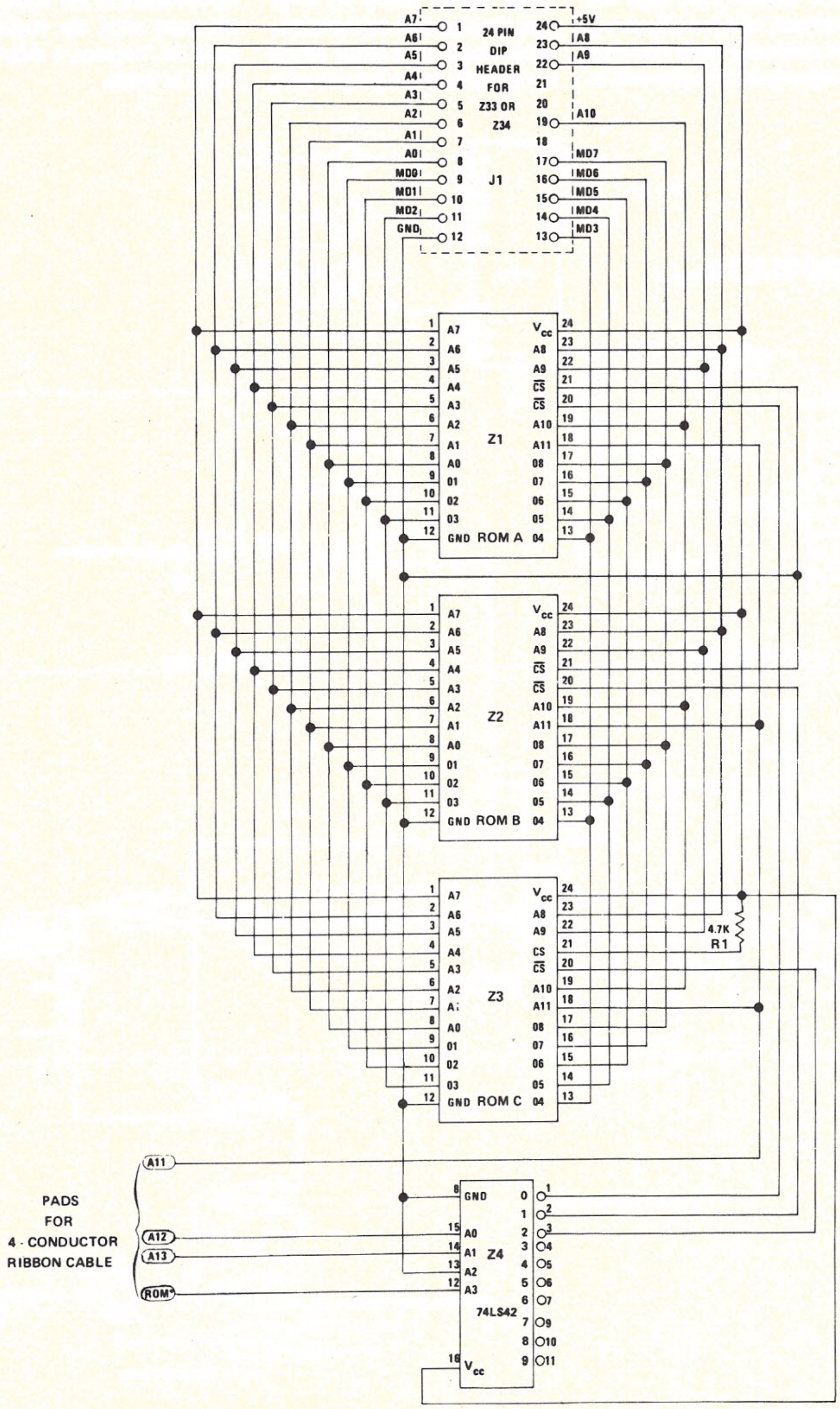


Fig. 5. Level II BASIC schematic. (Reprinted from the "TRS-80 Technical Reference Handbook," courtesy Radio Shack.)

The Big Moment

So, I tried it. The screen cleared, and a short message appeared in the upper-left corner. It said, "ξςξπτ√ υκ→ς?__." My computer was talking to me in Greek! There was obviously some incompatibility between the TRS-80 video driver and my TVT. The Level II manual tells me that the computer is supposed to say, "MEMORY SIZE?__." Anyway, I responded with a "32000," which appeared on the screen just as I typed it.

Hmmmm, my keyboard kludge was working alright and the numerals printed correctly, but the alphabet was in Greek! I hit the carriage return. Nothing happened for a moment, then another couple lines of Greek appeared.

You may be wondering where the Greek was coming from. Well, that is an easy one. The character generator ROM I bought for my TVT has Greek characters and some special math symbols where the control

UPI

PASCAL

FORTRAN

APL

STOCK QUOTES

TRS-80™ Super Version #1

Why limit yourself to Radio Shack® BASIC or spend hundreds of dollars on new programs and languages?

Through the phone lines, your TRS-80™ becomes a sophisticated and powerful computer. Access new languages, make vital decisions based on stock quote, earn money selling those top notch programs you write.

Two services, MicroNet™ and The Source™ open the world to the 80 user. Pick the group for your needs or both!!

MicroNet
- Hookup $9.00
- Per hour $5.00

The Source
- Hookup $100.00
- Per hour .... $ 2.75

These charges are billed to your Master Charge or VISA.

Houston Micro Computer Technologies, Inc. offers you three packages to choose from:

Package I (ST80III, Accoustic Modem, RS232C) with:
- MicroNet .... $460.
- The Source .... $540.
- Both .... $545.

Package II (ST80I, Accoustic Modem, RS232C) with:
- MicroNet .... $400.
- The Source .... $480.
- Both .... $485.

Package III for Mod II (HMCTERM, Accoustic Modem, Cable)
- MicroNet .... $500.
- The Source .... $580.
- Both .... $585.

ST-80 I ($25.00)
Set Serial Board from Keyboard
Auto Line Feed
Line Printer on

ST-80 II ($85.00)
Dump Memory to Disk
Allow Lower Case Letters
Transmit Auto Logon
Transmit Buffer to Serial: Out Port
Serial: Into Memory

ST-80 III ($150.00)
Help Command
Auto Answer Auto Originate
Switch Status
Warm Program Restart
Job Log

HMCTERM ($200.00)
Converts TRS-80 Model II to Complete Intelligent Terminal for Processing with Major Mainframes

To order by phone or for local dealer information call: **713/661-2005**
Texas residents add 6% sales tax • MasterCharge • Visa

HOUSTON MICRO-COMPUTER TECHNOLOGIES, INC.

Home and Business Computer Specialists
5313 BISSONNET • BELLAIRE • TEXAS • 77401 • 713/661-2005
Dealer inquiries invited.

33

characters would normally be. Most video drivers don't actually send control characters to the video RAM; rather, they decode them and take the appropriate action. For some strange reason, the TRS-80 video driver was changing the normal alphabetic codes to control codes before sending them to the video RAM.

The First Program (in Greek)

I know that some people think that programming computers is like talking in Greek, but this is ridiculous! The Level II manual has a short program in the back which will display all of the graphics characters. I typed the program into my computer . . . in Greek! I changed it slightly, so it would print all characters not including the control codes. After I finished typing it, I listed it. Since I can't read Greek, I couldn't tell if I had it right or not, but at least the list command worked.

Next I typed "τχο;" that's RUN, for those of you who don't know Greek. Characters flashed by on the screen, and scrolled off before I could read them. I ran it again, but I halted the computer before everything disappeared. The special characters and numerals looked good. Then there were two sets of Greek characters where the uppercase and lowercase should be. Next came the graphics characters, which looked all right.

Fig. 6. Level II TRS-80 memory map. (Reprinted from "Level II BASIC Reference Manual," courtesy Radio Shack.) I have added a few addresses I have discovered.

D/LEVEL II TRS-80 MEMORY MAP

ADDRESS DECIMAL	ADDRESS HEXIDECIMAL	
0	0000	
		LEVEL II BASIC ROM
12288	3000	
		RESERVED
14302	37DE	COMMUNICATION STATUS ADDRESS
14303	37DF	COMMUNICATION DATA ADDRESS
14304-7	37E0-3	INTERRUPT LATCH ADDRESS
14304-7	37E0-3	DISK DRIVE SELECT LATCH ADDRESS
14308-11	37E4-7	CASSETTE SELECT LATCH ADDRESS
14312-5	37E8-B	LINE PRINTER ADDRESS
1436-9	37EC-F	FLOPPY DISK CONTROLLER ADDRESS
14336	3800	
		TRS-80 KEYBOARD MEMORY
15360	3C00	
		TRS-80 CRT VIDEO MEMORY
16383	3FFF	
16384	4000	
		LEVEL II BASIC FIXED RAM
		VECTORS (RST'S 1 THROUGH 7)
16402	4012	
16405	4015	KEYBOARD DEVICE CONTROL BLOCK
		DCB + 0 = DCB TYPE
		+ 1 = DRIVER ADDRESS
		+ 2 = DRIVER ADDRESS
		+ 3 = ∅
		+ 4 = ∅
		+ 5 = ∅
		+ 6 = 'K'
		+ 7 = 'I'
16413	401D	VIDEO DISPLAY CONTROL BLOCK
		DCB + 0 = DCB TYPE
		+ 1 = DRIVER ADDRESS (LSB)
		+ 2 = DRIVER ADDRESS (MSB)
		+ 3 = CURSOR POS N (LSB)
		+ 4 = CURSOR POS N (MSB)
		+ 5 = CURSOR CHARACTER

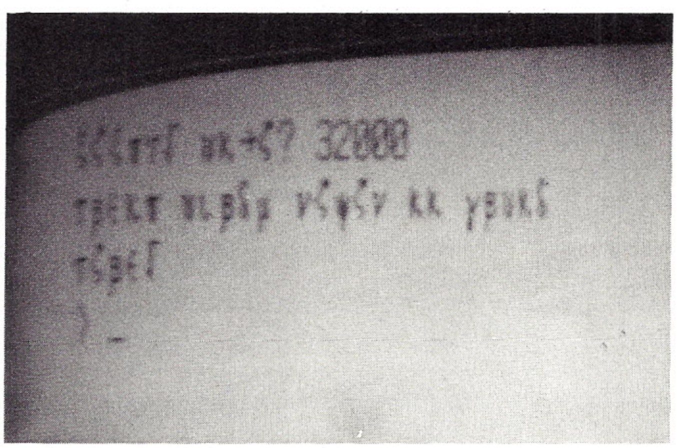


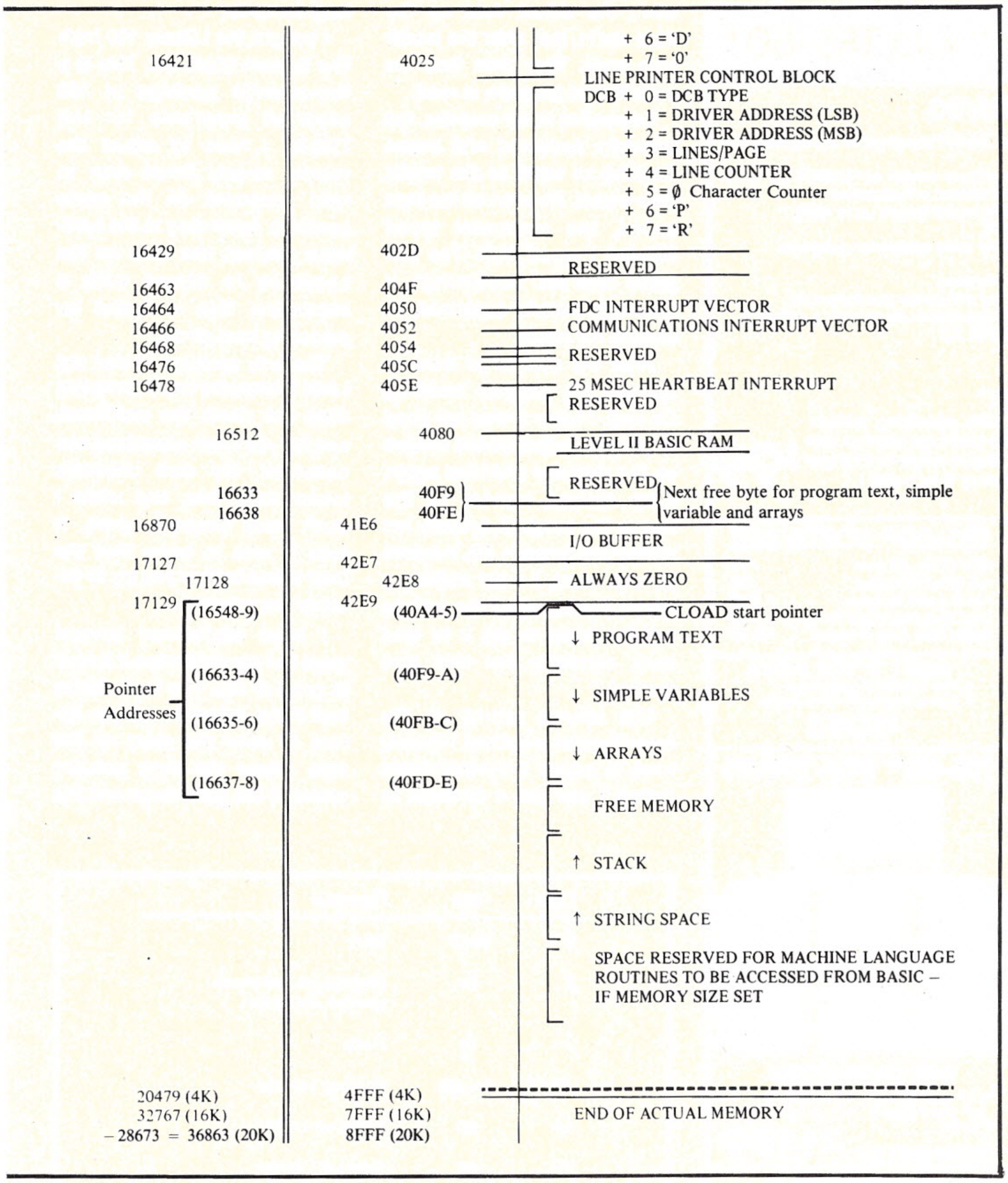
Photo 3. Initial run of Level II BASIC. Translation:
MEMORY SIZE? 32000
RADIO SHACK LEVEL II BASIC
READY
>_

Finally, there were all of those spaces, as everything scrolled off the screen. The Level II manual has a good explanation for the scrolling phenomenon. The codes, C0H to FFH, are space-compression codes for 0–63 spaces. So, by printing all of those codes, I had printed about 2000 spaces to the screen. I changed the program so it did not print the space-compression codes and ran it again. This time it didn't scroll off the screen.

Video Driver Patch

I remembered something I had seen in the Level II manual, which showed a memory map, which had a detailed description of some of the RAM locations used by the Level II BASIC. I was interested in a short section of 25 RAM locations containing three device control blocks. There were control blocks for the keyboard, the video display and the line printer. As you can see from Fig. 6, among other things, each block contains a driver address.

Now I figured all I had to do was to change the driver address to my own video driver, and I would be in business. I tried it. Nothing! I guessed that they used a different register to transfer the data byte. With this in mind, I set up a breakpoint at



the entry point of my video driver. I was then able to determine that the data was always in register C; my driver required the data in register A. I patched this in and tried again.

Now I was getting data, but everything was on the same line! There were only carriage returns and no line feeds! It seems the TRS-80 video driver automatically generates a line feed when it gets a carriage return. As it turns out, my video driver generates a carriage return if it gets a line feed! So I checked for carriage returns and converted them to the line feeds and tried again.

Now that was much better! Everything seemed to work. Well . . . almost everything. The clear screen function did not work. I know this used to work when everything was in Greek. Referring again to the Level II manual, I noticed they have a table that describes all of the control codes that are implemented (Table 1).

I had two choices: modify my video driver to handle all of the control codes or try to see if I could patch their video driver so it would work. Half out of curiosity as to what they were doing and why it worked (on a TRS-80) and half because I didn't really feel like rewriting my driver, I disassembled their driver.

As I had guessed earlier, they are converting both upper and lowercase letters to control codes. The question is, "Why do they do this and how come it works?" The answer is in the hardware manual. It seems they thought it would be less expensive to use only seven bits of information in the video RAM instead of eight. They use one bit to select graphics characters or regular characters. That leaves six bits for the ASCII code.

But the ASCII code is a seven-bit code; how can that work? They cheat a little. The seventh ASCII bit is generated with a NOR gate from two other bits. This means that if they sent an actual lowercase code to the video RAM, it would be printed as a numeral or a special character. So they had to convert lowercase to uppercase. It was probably simpler to convert both upper and lowercase letters to control codes than to just change lowercase to uppercase.

Anyway, as far as they were concerned, that particular bit didn't really matter because it was not even in the RAM! Personally, I think they should have spent the extra buck on one more memory chip, then they could have had both upper and lowercase on the computer.

The final solution I came up with was to duplicate the first dozen instructions of their driver and then skip over the section that screws up the characters and jump back to their driver. The total patch is about 40 bytes.

Listing 3 shows that I have included two more small patches to the driver. The first changes the up-arrow code from 5B (which prints a left bracket([)) to 1C, so it prints an up arrow on my TVT. Radio Shack mentions in the Level II manual that some TRS-80s may print the up-arrow as a left bracket. The second allows me to bypass the space-compression codes and print 64 more of my programmable characters instead. This is accomplished by poking one byte in a memory location.

The Cassette Interface

Having gotten the video driver working made me feel very confident. I was now ready to attack the cassette interface. I placed the Blackjack tape supplied with the Level II kit in the recorder (a Radio Shack CTR-40) and typed CLOAD. I have a small tape controller box, which enables me to hear the data while the computer is reading it. This is convenient because you can tell the difference in the sound of the actual data and the leader tone on the tape.

I turned on the recorder and hit the return key. One nice thing about the TRS-80 cassette driver is that two asterisks flash in the upper-right corner of the screen when the computer is reading data. The asterisks first appear

when the actual data on the tape starts, just after the leader tone ends. They then flash as each line of program is read.

Somewhat to my surprise, the asterisks appeared and began flashing as soon as the leader tone ended. As soon as the data ended, the computer typed READY. I typed RUN. The program started executing! It asked me several questions, including my name.

After my second or third response, the program bombed. Oh well, I knew it was too good to be true. I adjusted the volume on the recorder and tried again. After several repeats of the above, the program actually ran all the way through. Ah, success at last. Next, I tried making a tape. I had to adjust the volume several times to get it to read back correctly, but this also worked.

The volume setting on the tape recorder is critical. I usually have to adjust it several times before I can get a program to load correctly.

I bought the Library 100 from The Bottom Shelf, Inc. This is a five-cassette package of 100 assorted programs for the TRS-80. I have to adjust the volume several times even to read programs on the same cassette. According to the hardware manual, the data on the cassette is saved with a checksum. This is useful for detecting load errors.

The only problem is that the Level II cassette loader program does not check the checksum and tell you when a bad load has occurred. My own cassette loader does this, and while I don't have frequent errors, it sure is nice to know that the load is bad before you try to execute the program.

I have discovered several ways to help determine if a load is good or not. The load will be bad if the asterisks appear before or after the point on the tape where the data actually starts; if the data stops and the

Code	Hex	Function
0-7	00–07	None
8	08	Backspaces and erases current character
9	09	None
10-13	0A–0D	Carriage returns
14	0E	Turns on cursor
15	0F	Turns off cursor
16-22	10–16	None
23	17	Converts to 32 character mode
24	18	Backspace ← Cursor
25	19	Advance → Cursor
26	1A	Downward ↓ linefeed
27	1B	Upward ↑ linefeed
28	1C	Home, return cursor to display position(0,0)
29	1D	Move cursor to beginning of line
30	1E	Erases to the end of the line
31	1F	Clear to the end of the frame

Table 1. Control codes decoded by the video driver on the Level II BASIC ROMs. (Reprinted from "Level II BASIC Reference Manual," courtesy Radio Shack.) I have added hex codes.

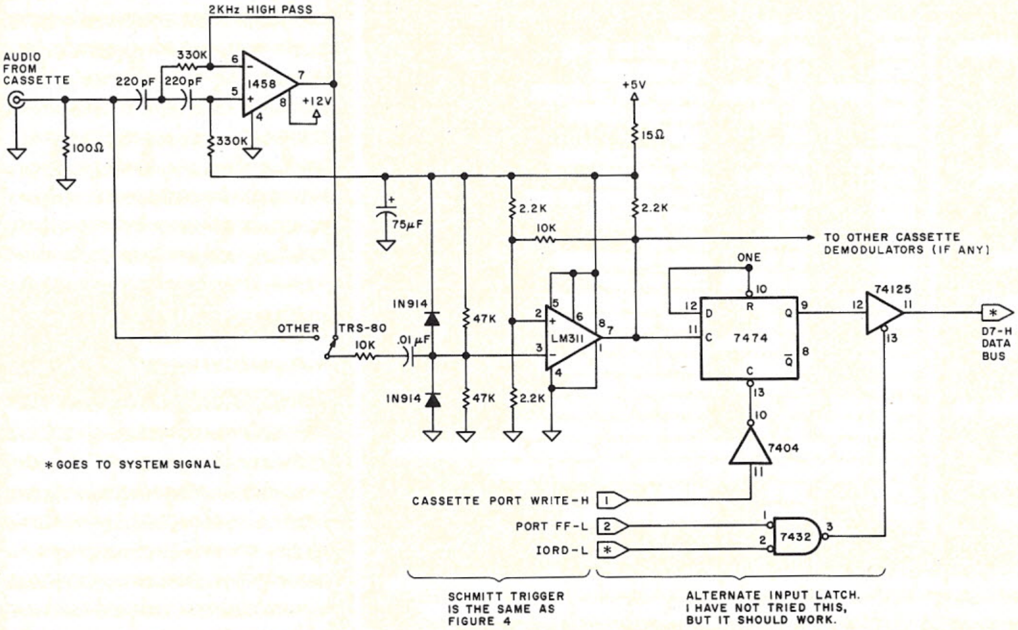


Fig. 7. Cassette input circuit I am now using. The 2 kHz high-pass filter is switched in to read Radio Shack tapes. The Schmitt trigger section is the same as in Fig. 4. The input latch is simpler than that shown in Fig. 4.

computer doesn't immediately respond with READY; if READY occurs before the data ends; or if the asterisks do not flash. If the asterisks flash slowly or erratically, the load may be bad. This clue takes some getting used to since the flash rate is not the same for all programs. You have to get a feel for how the asterisks normally flash.

If any of these symptoms occur, you will have to reload the program. Several of these problems cause the computer to hang up. A reset must then be issued to get back to BASIC.

During the next few weeks, I tried all of my 100 programs. I found that some of the tapes read fairly well, while others were very poor. These tapes have the same programs recorded on both sides as a backup. I found that I couldn't read some programs at all; I could read only one side correctly on some tapes; and I could read both copies on others. I tried reading some of these programs on a real TRS-80, and some that I couldn't read worked.

Since my input circuit was considerably simpler than the one they use, I breadboarded their circuit and tried it. It worked much better. The volume setting was less critical, but it was still more sensitive than I would have liked. With some experimenting, I found that I only needed the high-pass filter section of their interface. Since the TRS-80 tape format was so much improved with the filter, I tried it on my 2400 baud interface. It bombed. My interface became totally useless with the active filter.

The reason I attribute to this seeming inconsistency is that the Radio Shack recording method is an amplitude modulation scheme, while my interface is a phase modulation scheme. The active filter adds too much phase distortion for my interface to work properly.

The final circuit I implemented for my cassette interface is shown in Fig. 7. The switch is to select Radio Shack or other recording methods. I'm not really sure if my circuit is more or less reliable than Radio Shack's, but my circuit seems adequate. Most of the tapes read through with two or fewer volume adjustments. Some don't need any adjustments. I don't use my Radio Shack interface to save programs anyway, since my 2400 baud interface is nearly five times faster.

One feature of the Radio Shack cassette interface I haven't built is the motor control circuit. I've been using my cassette interface for a year and a half, and I don't think a motor control is necessary. I do use the motor control signal to change the clock frequency and to enable the output circuit though. This works very well.

Keyboard and Printer Patches

I decided to get rid of that keyboard kludge I was using. I wrote the short driver in Listing 4. This program simply checks the keyboard status bit and either returns a null if it is not set or returns the character. It also checks for and changes two characters that were different on my keyboard than what the

```
01080 ;VIDIO DRIVER PATCH - PRINTS UPPER AND LOWER CASE
01090 VIDPCH  LD L,(IX+3)   ;GET CURSOR POINTER
01100         LD H,(IX+4)   ;GET CURSOR POINTER
01110         JP C,049AH    ;I'M NOT SURE WHAT THIS IS
01120         LD A,(IX+5)   ;GET CURSOR CHARACTER
01130         OR A
01140         JR Z,PATCH1
01150         LD (HL),A
01160 PATCH1  LD A,C        ;GET CHARACTER
01170 ;THE FOLLOWING FEW LINES ADJUST THE UP ARROW CODE FROM THE
01180 ;TRS-80 CODE TO THE EQUIVALENT CODE ON THE CHARACTER GENERATOR I
01190 ;HAVE, WHICH IS MCM6571A.
01200         CP UPAROW     ;THIS IS THE UP ARROW CODE
01210         JR NZ,PATCH2  ;IS NOT UP ARROW
01220         INC C         ;YES ADJUST
01230         JP 0467H      ;DON'T BYPASS UPPER,LOWER ADJ
01240 PATCH2  CP ' '        ;CONTROL?
01250         JP C,0506H    ;YES, DO IT
01260         CP 80H        ;GRAPHICS?
01270         JP NC,VIDJMP  ;YES, DO IT
01280         JP 047DH      ;NO, ALL OTHER
```

Listing 3. Patch to the TRS-80 video driver eliminates the section that converts lowercase and uppercase character codes to control character codes. This permits both upper and lowercase to be printed.

TRS-80 Key	ASCII	Hex	Normal Keyboard
BREAK	SOH	01	CTRL A
←	BKSP	08	CTRL H
→	HT	09	CTRL I
↓	LF	0A	CTRL J
↑	[	5B	[
ENTER	CR	0D	RETURN
SHIFT ←	CAN	18	CTRL X
SHIFT →	EM	19	CTRL Y
SHIFT ↓	SUB	1A	CTRL Z
SHIFT ↑	ESC	1B	ESCAPE
CLEAR	VS	1F	

Table 2. Control codes generated by the keyboard driver on the Level II BASIC ROMs. Your keyboard must generate these characters also.

```
01290 ;TRS-80 SUBSTITUTE KEYBOARD DRIVER
01300 KBDSUB  LD HL,KBDST  ;KEYBOARD STATUS ADDRESS
01310         BIT 0,(HL)
01320         JR NZ,KBSB1  ;YES, ITS READY
01330         XOR A        ;NOT READY
01340         RET          ;RETURN WITH A NULL
01350 KBSB1   RES 0,(HL)   ;RESET STATUS BIT
01360         INC HL       ;BUMP POINTER TO DATA
01370         LD A,(HL)    ;GET DATA
01380         CP RUBOUT    ;IS IT A RUBOUT?
01390         JR NZ,KBSB2  ;NO
01400         LD A,BS      ;YES, LOAD BACKSPACE
01410         RET          ;AND RETURN
01420 KBSB2   CP ENQ       ;THIS IS FOR CLEAR KEY
01430         RET NZ       ;NOT CLEAR, RETURN
01440         LD A,1FH     ;YES, LOAD CLEAR CODE
01450         RET          ;AND RETURN
01460 ;PRINTER DRIVER PATCH
01470 TRSPRT  LD A,C       ;MOVE DATA TO CORRECT REG
01480         LD HL,CHRCNT ;GET POINTER
01490         CP 0CH       ;FORM FEED
01500         JR Z,FORMF   ;INITIALIZE LINE COUNT
01510         CP 0DH       ;CARRIAGE RETURN
01520         JR Z,CR      ;DO CARRIAGE RETURN
01530         LD A,(HL)    ;GET CHAR COUNT
01540         CP 40H       ;=64?
01550         CALL Z,CR    ;INSERT A CR IF =64
01560         LD A,C       ;GET DATA AGAIN
01570 TRSPR1  CALL TYPOUT  ;OUTPUT CHAR TO PRINTER
01580         INC (HL)     ;BUMP CHAR COUNTER
01590         RET          ;DONE
01600 CR      DEC HL       ;POINT TO LINE COUNT
01610         LD A,(HL)    ;GET LINE COUNT
01620         DEC HL       ;POINT TO LINES PER PAGE
01630         INC A        ;BUMP LINE COUNT
01640         CP (HL)      ;AT LIMIT?
01650         INC HL       ;POINT TO LINE COUNT
01660         JR Z,NXTPAG  ;YES, AT LIMIT
01670 CR1     LD (HL),A    ;SAVE NEW LINE COUNT
01680         INC HL       ;POINT TO CHAR COUNT
01690         LD (HL),0FFH ;RESET CHAR COUNT
01700         LD A,0DH     ;GET A CARRIAGE RETURN
01710         JR TRSPR1    ;DO CARRIAGE RETURN
01720 FORMF   DEC HL       ;POINT TO LINE COUNT
01730 NXTPAG  CALL OK      ;THIS ROUTINE PRINTS "OK?"
01740 ;AND WAITS FOR A CHARACTER FROM THE KEYBOARD BEFORE CONTINUING.
01750 ;THIS ALLOWS ME TO PUT ANOTHER PAGE IN MY PRINTER.
01760         XOR A        ;CLEAR A
01770         JR CR1       ;FINISH BY DOING CR
```

Listing 4. Keyboard driver replaces the one on the Level II ROM. It works in conjunction with my normal keyboard interrupt routine in Listing 1. The printer patch replaces the Level II printer driver.

ROM expected.

Actually, there were more characters that didn't quite match, but I reprogrammed my keyboard encoder EPROM to fix those. The reason I didn't fix all of them on the EPROM is because I would have had to change my monitor that used those characters. I figured it was better this way. Table 2 shows the control characters generated by the TRS-80 keyboard.

The printer patch adds a few features that my driver didn't have but are assumed by the Level II ROMs. The major feature is to add extra carriage returns when a line exceeds 64 characters in length. My first printer patch did not do this, and when I listed BASIC programs that had multiple statement lines longer than 64 characters, the extra characters would not print. I also added a lines-per-page counter. When the line count is at the limit, the program waits for me to put another page in my printer.

The routine TYPOUT in Listing 4 converts the ASCII input to the special code required by my modified Olivetti Lexikon 82 typewriter. The form-feed check in the program permits resetting the line counter to zero. This should be used before any new listing, so it starts at the top of a new page. From BASIC, the following line will work: LPRINT CHR$(12);.

Specific Hardware Requirements

The single most important hardware requirement for your system is the use of interrupts. The reason for this is the way the keyboard is set up. When the Level II ROM initializes its RAM space, it assumes that you have a memory-mapped keyboard. If you don't have a memory-mapped keyboard and you don't have interrupts either, there is no way you can talk to the Level II program, and the computer will be hung up.

The only way to get the pro-

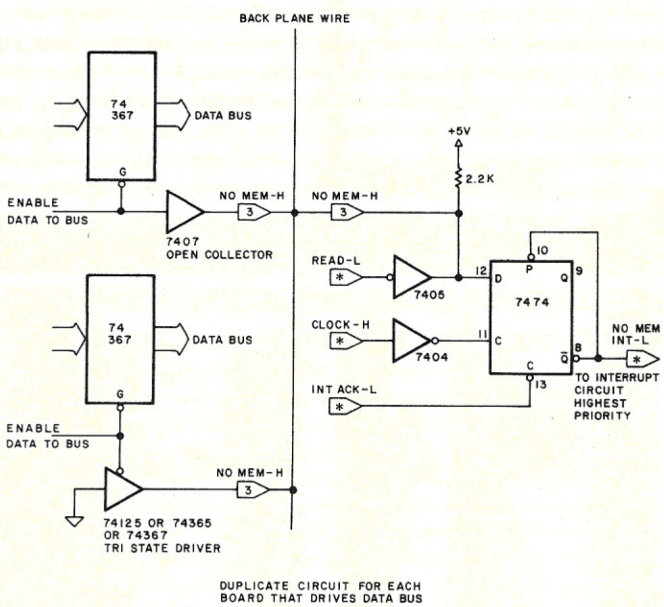


Fig. 8. To use this "No Memory" interrupt circuit, you must add a gate to every board in your system that drives the data bus. This circuit will generate an interrupt every time a nonexistent memory address or input port is read. This should be the highest priority interrupt in your system.

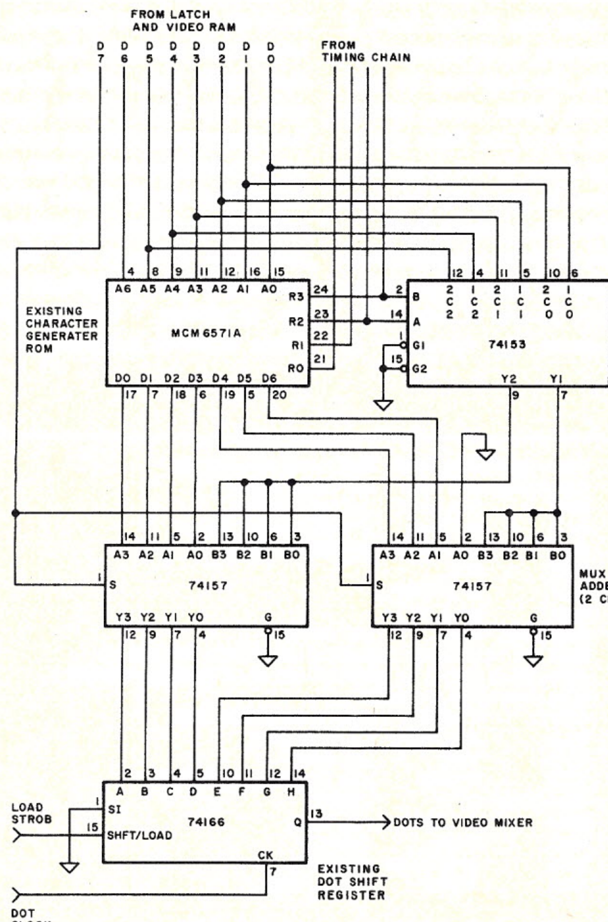


Fig. 9. This simple graphics mod will work for TVTs that use the MCM6571A character generator ROM. If you have a different character generator, the pin numbers will be different, but the circuit will be similar.

gram's attention is with an interrupt. If you have an interrupt-driven keyboard, you could use a program such as Listing 1 to simulate the TRS-80 memory-mapped keyboard, as I did at first. Otherwise, you need some other means of interrupting the computer. This could be as simple as a switch to the interrupt line on the Z-80. The interrupt service routine could simply change the keyboard driver address and then return to the Level II program.

There are only two situations where you could get by without any interrupts. If you actually connect your keyboard the same way as Radio Shack did, you wouldn't need interrupts. If you already have a keyboard connected some other way, rewiring it is probably undesirable. Or, if you have a hardware front panel, you could interrupt the computer that way and change the keyboard driver address. While that is not really very difficult, it is kind of a bother to flip all those switches. My system includes a front panel, and I didn't want to do it that way.

The method I used to interrupt the computer is a bit unusual for a microprocessor. I have a circuit in my computer that generates an interrupt if the computer attempts to read a memory address at which there is no memory installed (see Fig. 8). This interrupt saves all the registers, prints a "No Memory" message and jumps to my monitor. When the ROM tries to read the keyboard, this interrupt is generated because I don't have any memory there. From here I simply type BC, a monitor command that stands for BASIC Continue.

Listing 2 is the program. Its function is very simple—it merely sets up the new driver addresses for the keyboard, TVT and the printer. Then it restores all the registers and returns to where it was interrupted.

TVT Specifics

If your TVT is a memory-mapped device with 16 lines of 64 characters, you should have no problems getting it to work with Level II BASIC. You will have to change its address to 3C00-3FFF. If you don't have a programmable character generator, you will have to modify the TVT to implement the TRS-80 graphics. The modification should consist of only three ICs as shown in Fig. 9.

Fig. 10 shows the graphics-character format. As you can see, each character cell is divided into six blocks. Each block is controlled by one bit in the video memory. The most significant bit determines if a particular character is a graphics character or a regular character. The multiplexers simply steer the bits to the appropriate positions.

This circuit will work for TVTs, which have a character cell con-

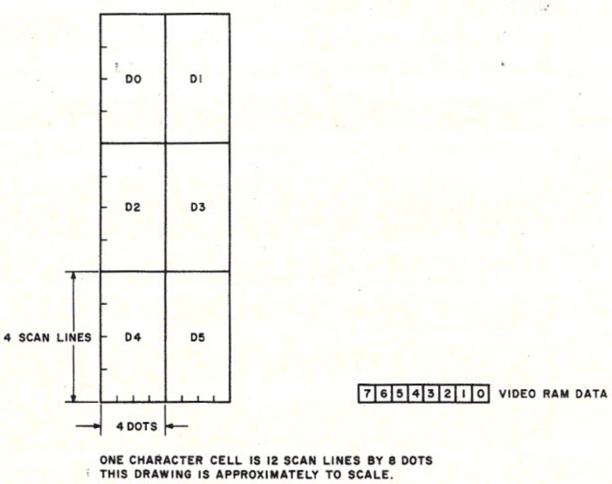


Fig. 10. Scale drawing of one character cell shows that each graphics dot is approximately twice as tall as it is wide. The video RAM bits that control each graphics dot are also shown. This format matches the circuit in Fig. 9.

sisting of 12 lines of eight dots. If your TVT has 12 lines of six dots, simply tie the two outputs from mux A to each of three inputs on mux B instead of the four shown. If your TVT has a different arrangement of lines and dots, you have several choices.

First, you could stretch or shrink some of the graphics dots so they fill the available lines and dots in the character cell. This may cause some graphics dots to be different sizes than other ones if the total number of lines and dots are not evenly divisible by three and two, respectively.

Second, you could modify your TVT so it has a line count divisible by three and a dot count divisible by two. This is a bit tricky and should be attempted only after you have examined the schematic and understand the timing details of the TVT. The first mod is simpler and doesn't affect the timing, but you should still closely examine the schematic of your TVT before attempting to install the change.

Third, you could forget about the graphics. This is the simplest solution, but since a lot of game programs use the graphics, you may not want to do this. If you never play games, then you don't need the graphics anyway.

I suggest you try the first solution before trying the second. The slightly different size dots will go unnoticed in many applications anyway. My own TVT has a software-selectable character cell size. I can select 13 by 9 or 12 by 8. I normally operate in the 13 by 9 mode and have found it satisfactory in many graphics applications.

If your video terminal is a completely separate unit from your computer, you obviously don't have a memory-mapped device. This means you can't use any part of the TRS-80 video driver. You will have to either write your own or modify the one you are presently using. The most important thing is to have the control characters respond correctly (see Table 1).

There are a few features in Level II BASIC that won't work with this type of setup. The graphics functions, SET, RESET and POINT, won't work, although you could send the graphics characters to the terminal like any other character. The PRINT@ and POS commands won't work either. Everything else should be fine though.

Your First Run

When you first try to run the Level II BASIC, you may have a different sequence of events than I do, depending on just how your hardware is configured. As you recall, my first run produced Greek characters. I no longer get Greek when I initialize the BASIC ROM. The first thing that appears is a "No Memory" message. This occurs when the ROM attempts to read the keyboard memory. I then type BC (BASIC Continue).

As described earlier, this changes some of the RAM locations just initialized by the ROM and returns to Level II BASIC. From here, my system behaves just like a TRS-80.

If you don't have a "No Memory" interrupt on your system, and depending on what your TVT does with control characters, your system could produce Greek characters, some strange graphics characters or absolutely nothing. The next display will depend on what you have in the keyboard memory area. If this memory is all zeros, you will only see one line of whatever characters your system is producing. If the memory is all ones (FF hex) or random data, you should see several lines of these characters continously being written to the TVT and scrolling off the screen.

No matter what you see, you should now hit your interrupt button (control-Z, or whatever) to put you back into monitor. After typing the BASIC Continue command, you should have a blank screen.

The ROM is now waiting for your response to the MEMORY SIZE question, even though you can't see that message. Typing anything should cause it to appear on the screen. Since there may be several unknown characters in the keyboard buffer, you should first delete these with the back-arrow key. When the cursor stops moving back, all characters have been deleted. Now answer the MEMORY SIZE question as you wish. If you hit a carriage return with garbage data, the ROM will ask the MEMORY SIZE question again.

One final note: if, on your system, memory address 37ECH returns anything other than 00 or FFH when read, the ROM may attempt to boot the disk. I'm not sure exactly what will happen, but it will most likely get hung up and do nothing. If you have no memory at that address, you should be OK, since most systems read FFH or 00 to nonexistent memory.

Conclusions

For someone with a Z-80 microcomputer system who is looking for a good BASIC and would prefer to have it on ROM, Radio Shack's Level II ROM add-on kit for their TRS-80 is a good way to go. The price is reasonable—less than many BASICs that only come on cassette. If you consider the additional cost of EPROMs to put another BASIC on ROM, the Level II BASIC is less expensive than any other I know.

That the TRS-80 is the most popular microcomputer today ensures that there will be more directly compatible software than any one person can use. The ROM also contains a floppy disk bootstrap routine. This allows easy addition of one or more mini-floppy disk drives for a more versatile system. Radio Shack's TRSDOS may not be the best, but at only $14.95, it certainly is the most inexpensive disk operating system I have ever seen. ■

References

"TRS-80 Microcomputer Technical Reference Handbook," Radio Shack.

"LEVEL II BASIC Reference Manual," Radio Shack.

"TRSDOS & DISK BASIC Reference Manual," Radio Shack.

And now the news...

CompuServe announces the availability of international, national and regional news through your personal computer or terminal.

This up-to-the-minute service includes:
- late breaking international and national news, including features and syndicated columns
- national sports with tabular standings and box scores
- national business and financial news, including the latest stock prices and closings
- national broadcast wire
- Washington news wire
- weather updates
- and selected regional wires

Find out how your favorite football team fared or get a full account of the game in seconds. Find out how the market did today or what movies got good reviews. Check election results, key legislative votes, the latest on foreign affairs. Even background stories or predictions for the future. International events. National news. What diplomat is throwing a party in Washington.

All this and more is yours today through CompuServe.

A 300-baud modem is all your personal computer or terminal needs to access all the services available via local phone calls in more than 175 North American cities from 6 PM to 5 AM weekdays, weekends and most holidays. And the basic charge is only $5.00 an hour, billed in minute increments, to your charge card. Want more information? Write.

MicroNET™

CompuServe ✓147

Dept: K
Personal Computing Division
5000 Arlington Centre Blvd.
Columbus, Ohio 43220

64K Memory For the H8

Last month's article continues with construction and checkout of the memory.

Myron J. Seibold
PO Box 5131
Santa Ana, CA 92704

Last month, part 1 described the actual design of a 64K, single card semiconductor memory for the Heathkit H8 computer. Now let's look at its construction and operation.

Power Supply Circuits

The three power supply voltages required by the memory are easily obtained from the H8 computer. The unregulated H8 voltages, −18, +8 and +18, are converted into −5, +5 and +12 volts, respectively, using standard three-terminal voltage regulator chips (see Fig. 1).

The power supply current requirements are small: −5 volts at 10 mA, +5 volts at 1.0 amperes and +12 volts at 150 mA. The +5 volts should be supplied using four regulators driving separate loads, none of which should draw more than 500 mA. The integrated circuits should be split into four groups, each connected to a separate +5 volt power supply regulator. This is done to permit mounting the regulators on a small heat sink. The memory chips will constitute one of these four groups.

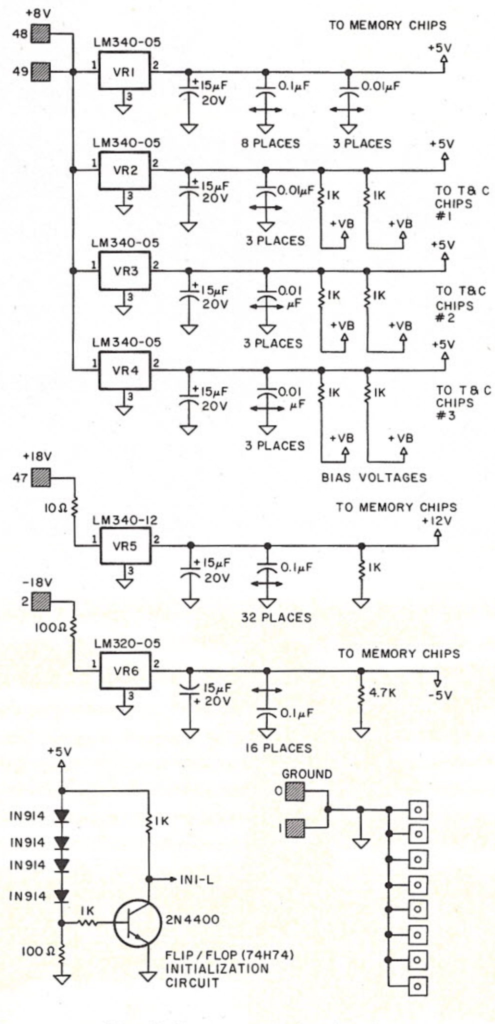


Fig. 1. Power supply circuits.

The average (dc) currents supplied by the +12 volt and −5 volt regulators are quite small. The *peak* currents these voltages must supply on the memory board, however, are large and are many times greater than the dc currents in both cases.

The peak current demand is supplied by the filter capacitors in the memory chip array. A large amount of distributed capacitance is required for this purpose. You must carefully follow the layout rules for the memory chip array. These currents must be supplied when needed, without inducing spurious signals. During refresh, these currents are drawn by all of the memory chips simultaneously.

Memory Chip Layout Rules

The wiring board layout requirements for dynamic memory chips are exacting. The memory chips themselves must be arranged together in a compact group. All three power supply voltages, as well as ground, should be respectively cross-connected at each memory chip (see Fig. 2). A pair of high-quality ceramic power supply filter capacitors should also be installed with each memory chip.

The memory chip ground connections should form a net with a memory chip at each connected intersection. The same is true for connections to +12 volts and −5 volts. The +5 voltage should be connected in a similar manner, but is not as critical and is used only to provide TTL compatibility. The +5 voltage is not required to retain data in the memory chips.

High-quality ceramic capacitors are essential to filter the

memory chip power supply voltages. Capacitor quality is directly related to price. Inexpensive capacitors will not work for this purpose. A capacitance of 0.1 uF should be used for each capacitor. The two capacitors should be located as close to each memory chip as possible. The CK05BX104K capacitor is suitable for this purpose and is recommended for its small size.

The other memory circuits should be laid out in a similar manner, although the layout requirements are not as strict. Only a single power supply voltage is used, and only one filter capacitor need be used for every three or four integrated circuit chips. A good ground structure, however, is always important.

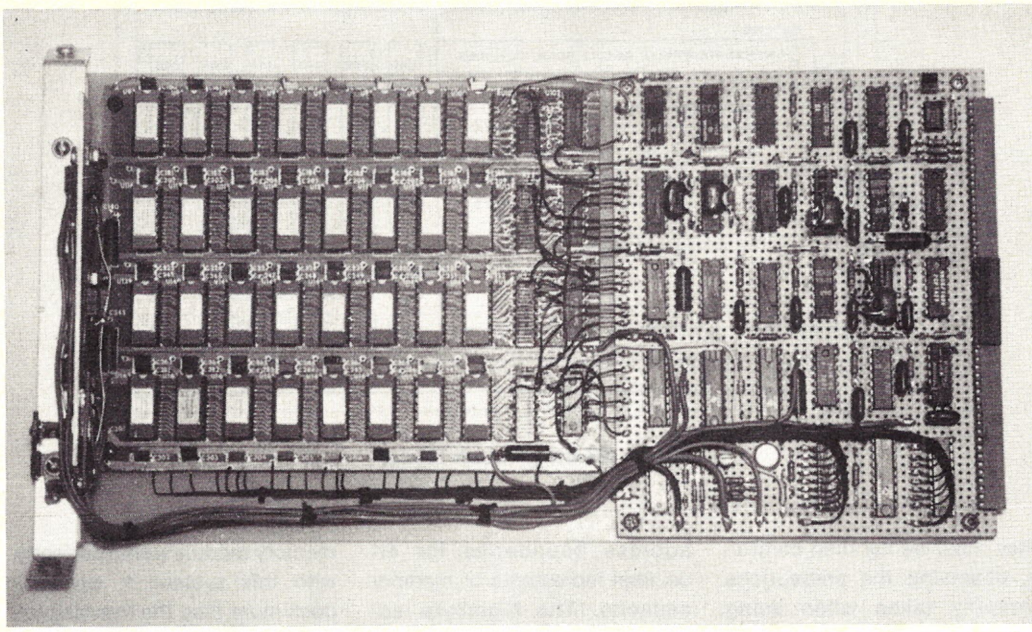


Photo 1. Full view of the memory module. Note light reflected from the array of 32 memory chips. All information is stored in these chips. The memory chips themselves occupy less than one half of the memory module. The remaining space is occupied by the memory chip support circuitry. This is characteristic of most dynamic memory designs.

Assembling the Memory Module

This memory was physically assembled in three parts on a retaining backboard identical in size to the printed circuit boards used in the H8 computer (see the memory layout drawing, Fig. 3). The memory chips and their address drivers were easily mounted on a printed circuit board cut from a surplus commercial memory board. I have a limited supply of surplus memory boards of different types from which suitable memory chip arrays may be cut. These printed circuit board pieces typically hold all 32 memory chips, about 70 filter capacitors, four resistor modules (or 32 1/8 Watt resistors) and four integrated circuit address and control driver chips. Using these boards is convenient and ensures that the memory chip array is properly wired and filtered.

The memory chips should be installed on these boards in high-quality sockets. These sockets will facilitate troubleshooting and memory chip replacement. The memory chip array can also be constructed on regular wiring board, such as Vector board, following the layout rules previously given. Again, sockets should be provided for the memory chips.

The remaining circuits were separately wired on a small piece of Vector board. The Vector board hole spacing is 1/10 inch in a square array. Small-size Vector pins are used with this board. Thirty integrated circuits can be laid out in a 6 × 5 chip array with ample space around them for discrete components.

Wire-wrap sockets are recommended for the integrated circuits. These sockets can be glued to the Vector board and the leads cut to a convenient length for wire (solder) connections. Either ¼ Watt or 1/8 Watt resistors may be used. The timing capacitors normally have a rating of at least 500 volts. Photos 1 and 2 of the memory module may differ slightly from the layout shown in Fig. 3. The layout is small, and component placement is not critical. Wiring connections should always be as short as possible.

Wires must be soldered to the memory chip printed circuit board for connection to the circuits on the Vector board. A few ground wires should always accompany any groups of signal wires. From the bottom of the memory chip array, for example, a bundle of eight data input lines, eight data output lines and four ground lines was brought out (see Photos 1 and 2).

The memory chip array printed circuit board and support circuit Vector board were mounted to the backboard using short standoff spacers. The two boards were then wired together. The voltage regulators were mounted to a standard Heath metal-mounting bail at the edge of the backboard. This bail acts as a heat sink for the voltage regulators. It is a poor heat sink, and four +5 volt regulators were used to reduce the regulator junction temperatures. The bail gets hot and may reach 150 degrees Fahrenheit (65 degrees Celsius).

The single +12 volt regulator can also be mounted on the bail. Silicon grease should be used when mounting the regulator chips. The −5 volt regulator does not get hot and need not be mounted on the heat sink. The bus connectors should be mounted directly on the Vector board. A pair of standard Heath gold-plated connectors should be used with a tie bar. The connectors should be well secured to the Vector board. Cut-down Vector pins can be used as solder eyelets for this purpose.

The memory can be assembled and operated in smaller versions than 64K. In this case, you can build a complete 64K memory with memory chips installed in only one or two rows of sockets. The memory will work with a single row of eight memory chips. This should not entail much expense and will provide 16K of memory capacity. Memory capacity can then be increased in 16K increments up to 56K by simply installing additional rows of memory chips.

A note of caution: the memory chips are MOS integrated circuits and are susceptible to damage by static electricity.

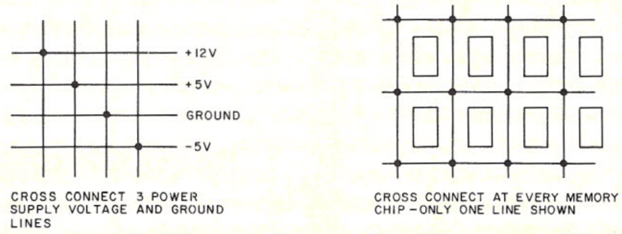


Fig. 2. Memory chip power-ground cross connections.

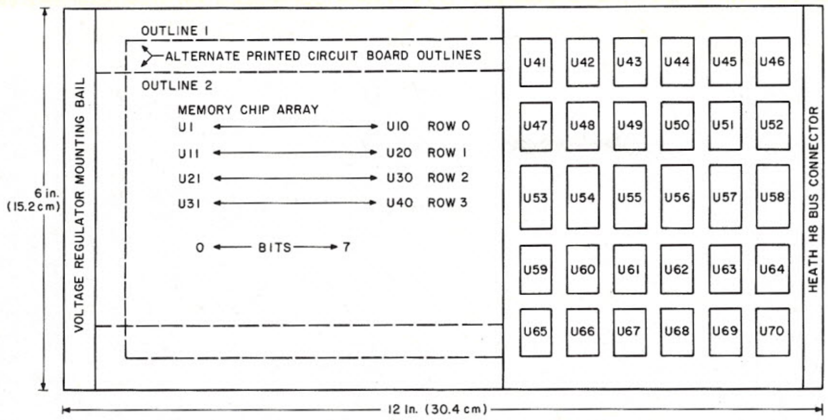


Fig. 3. Memory module layout.

They must be handled carefully, observing the precautions normally taken when using MOS devices. The memory chips must be handled one at a time and should be stored on a conductive foam pad when removed from the memory module.

Operation

When the memory is operating properly, the front panel (monitor program) can be used. It should be possible to load and store information anywhere in the memory address range from 8K to 64K. It is a heady experience loading and retrieving data in the higher address locations for the first time! All the effort in building and debugging the memory now seems worthwhile!

The H8 memory test routine should now be entered and run to ensure that the entire memory is functional and that there are no defective memory chips. The Heath H8 memory test routine is listed on pages 61 and 62 of the H8 operating manual. It is also listed and described in detail on pages 9 through 14 of the operating manual.

To test 56K of memory, the data placed in address location 040 105 (split octal notation) of the test routine should be changed from 057 to 377. This raises the upper memory test limit from 12K to 64K. (The memory test always begins at 8K.) To test smaller amounts of memory, the upper memory limit can be varied accordingly. See page 0-58 of the H8 software reference manual for the high-byte address boundaries for 4K decimal increments of memory capacity. The high-byte addresses are octal numbers and should be decremented by one (except in the case of 377) to set the memory boundary. For example, 200 – 1 = 177 sets the upper memory boundary at 32K, for a memory capacity of 24K.

The memory test begins at address 040 160, rather than at the 8K boundary of 040 000. Also, the memory test ends at xxx 260, rather than the upper limit of memory at xxx 377. These small amounts are excluded from the test because they are needed to operate the front-panel monitor program, as well as administer the test itself.

If the memory test runs successfully, it can be assumed that these excluded locations are also good. However, these locations can be thoroughly tested by interchanging RAS lines on the memory module. This is easy to do. RAS 0 and RAS 3 are interchanged with RAS 1 and RAS 2, respectively. This physically changes the memory chip rows responding to given address inputs in 16K blocks.

The memory test routine is primarily used to locate defective memory chips and certain kinds of problems in the memory chip array wiring. If there are problems in the memory chip timing and control circuits, the front-panel monitor program will not run.

I have a Heathkit H8 computer, H8-5 interface module, H9 video terminal and a pair of cassette units. The new 64K memory module works perfectly with this system. It certainly does more than the Heathkit 8K H8-1 static memory module it replaces. The computer can now be filled with interface boards, rather than with memory modules. This should eliminate any future need for a computer expansion chassis.

This memory has also been tested in an H8 computer using the H17 floppy disk system. The memory works perfectly with the floppy disk, and the large capacity of the memory effectively eliminates the need for a second floppy disk drive unit for disk copying. Single drive disk copying proceeds very rapidly when using a large memory.

Troubleshooting the Memory Module

This design is proven. Therefore, you should encounter a minimum number of problems in getting the memory to work—at least with a Heathkit H8 computer. In some cases, what appear to be memory problems may actually be problems in the computer hardware or software. It is now well established that the H8 is somewhat unreliable. The sockets on the voltage regulator leads on every H8 module should be removed, and these leads should be soldered directly to the voltage regulator terminals. It would also be worthwhile to replace the tin-plated motherboard connectors with gold-plated connectors of the same type.

Socket and connector problems occur unpredictably at infrequent intervals. Individual sockets can be tested by physical manipulation while running a program. Programs should run without interruption while sockets, or printed circuit boards, are flexed or tapped. A vibrator (use an engraving tool with a plastic ball on the point) set at low frequency can be very useful in this case.

Problems can also exist in the computer software. When running Heath cassette BASIC, you must software-set the upper memory limit below 40K. You can use the software configuration option to do this. Alternately, the RAS-3 line to the last row of memory chips can be lifted and the memory run as a 40K version. The RAS-3 memory chip line should be connected to a +5 volt bias resistor in this case.

Be very careful not to short connector pin 2 and 3 together on the bus when making measurements. This will destroy a number of integrated circuits on several computer modules. When wiring the memory module connector, you should note that the first pin is 0, not 1. If the connector is miswired, –18 volts will be placed on pin 3, with the above-mentioned consequences.

If the front-panel monitor program won't run, there is a problem in the memory chip timing and control circuits. Check all six power supply voltages first (+5 volts from each of the four regulators, +12 volts and –5 volts). An oscilloscope should then be used to check circuit operation. A dual-trace or external sync oscilloscope is required to make differential timing measurements.

Frequently, you can provide signals to the memory interface to check memory circuit operation when the front panel does not operate properly. Repeated pushing of front-panel buttons (especially reset) should set this up. You should first check all timing adjustments wherever possible (see timing specification, Fig. 4, part 1). Then you can check circuit operation for wiring errors and defective parts. It may be helpful to interchange RAS lines between rows of memory chips to

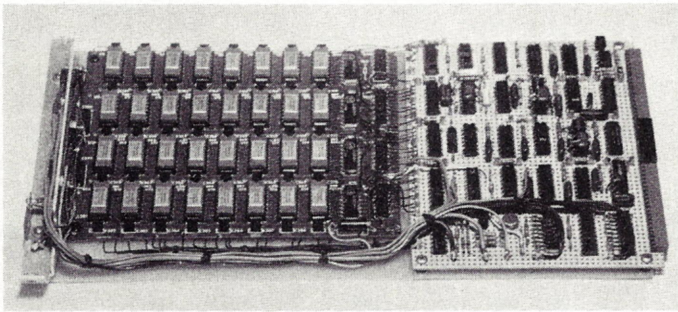


Photo 2. Another view of the memory module. Note use of a backboard to mount the separate memory chip and control circuit boards. A standard H8 bus connector and chassis mounting bail are used. The connector mounts on the control circuit board. The metal bail is mounted on the backboard.

locate or bypass rows with bad chips.

When adapting this design to another computer interface, you may have to devise and perform circuit modification experiments in order to obtain information. This is often necessary to locate problems that cannot be found using the oscilloscope. Interfacing involves both hardware adaptations to the computer bus and possible changes in the memory timing.

If you suspect a memory-read-access-time problem, RDYIN can be asserted for every memory cycle to check this. To accomplish this connect U58-1 to a +5 volt bias resistor or to U58-2. If the memory can be made to work with this modification—but not otherwise—the memory access time may be excessive.

To reduce access time use faster memory chips. The 4116-2 is 50 nanoseconds faster than the 4116-3 (see Table 1, part 1). Reducing the respective delays for address multiplexing and the assertion of CAS is required to take advantage of the faster memory chips. This should be done if the 4116-2 memory chip is used. Each of these delays can be reduced by one half when using the 4116-2 memory chip.

A memory-access-time problem can be solved, of course, by simply asserting RDYIN for every memory cycle. Using RDYIN with each memory cycle, however, is undesirable, as this will cause the computer to run more slowly. I experienced no problem with access time using the H8 computer with either the 4116-3 or the 4116-4 memory chips. I used the 4116-3 to provide an adjustable margin of reliability.

The refresh cycle must terminate before a computer memory cycle can begin. Access time will be adversely affected if the refresh cycle delays the beginning of a computer memory cycle when refreshing in the transparent mode. The input at U54-9 must go high before the input at U54-10 goes high to begin a computer memory cycle, in this case. This is ensured by the 1400 nanosecond maximum specification (Fig. 2, part 1).

Once the front panel monitor program is running, you can use the Heath H8 memory test to test the memory chips. This test will locate about 90 percent of all memory chip problems. No memory test can locate *all* memory problems. Dynamic memories are also susceptible to soft (nonrepeatable) errors caused by alpha particle radiation from the memory chip case materials. Fortunately, these errors are usually infrequent. ■

Heathkit H8 owners who would like to build their own memories can purchase printed circuit boards from the author. Fully assembled and kit versions of this memory in several different (depopulated) memory sizes are available.

★★

SURPLUS "SELECTRIC" SPECIAL!

"SELECTRIC" TYPEWRITER TERMINAL

Just imagine; an IBM Model 725 "SELECTRIC" typewriter built into a complete table-top RS-232 terminal! These **surplus** terminals were formerly on lease and appear to be in good condition (we test 'em to make sure the *printer is functional!*) These fantastic **BCD-Coded** terminals feature:

- 15" CARRIAGE
- 725 'SELECTRIC'
- RS-232 I/O
- 132 COLUMNS
- Sim. to IBM 2741
- Std. Typewriter Kbd.
- MAX: 15 CPS RATE
- 10 Chars./Inch
- Removeable Type Sphere
- 134.5 BAUD I/O
- 88 Character Set
- 6 Bit BCD CODE
- Attractive Case
- Upper/Lower SHIFT

ONLY **$469.00 Ea.!**

While we will check out each unit, we MUST offer these unique bargains *"AS-IS"*: Meaning they may need some service but are basically operational. Add $20.00 for packing crate, you pay shipping on delivery.

ALSO INCLUDES: Type ball, I/O circuit boards, power supply & some data. Sorry, no power cord included.

—SPECIAL OFFER!!—

Buy 2, take **20% Off** the Full Price—
You Pay Only .... **2 for $750.00**

"SELECTRIC"* PRINTER MAINTAINANCE MANUAL

JUST IN!! We now have available some excellent printer maintainance manuals. These are the most thorough manuals we've seen. Well worth the price! .... *ONLY* **$25.00ea.**

* *"SELECTRIC" is an IBM Trademark*

✓110

CFR Associates, Inc.

MAIL ADDRESS: P.O. Box 144 NEWTON, N.H. 03858

WAREHOUSE: 18 GRANITE STREET HAVERHILL, MASS. 01830

(617)372-8536

Phone Orders Are Welcome

Here's a better way to create/edit your North Star basic programs

N★BUS has a unique interactive GLOBAL LINE EDITOR, with powerful source preparation/editing features, unmatched anywhere at ANY PRICE!

- ✓ **Interfaces with ANY release 4 or later BASIC by RUNning a supplied program.**
- ✓ **Co-resident with BASIC — merely type the command "EDIT" without a line number.**
- ✓ **All N★BUS programs are assembly language fast!**
- ✓ **Line scrolling plus these exciting commands:**

EDIT MODE • INPUT MODE • BOTTOM • TOP • BASIC • BYE • APPEND • NEXT • PRINT • COPY (lines) • DELETE (lines) • SCALE • PROMPT • CHANGE • CHANGE (global) • ERASE COLUMNS • LOCATE • LOCATE (global) • TAB • TCHAR • WINDOW • WINDOW NEXT • WINDOW PREVIOUS • VIEW • INSERT • GO (to a line) • MOVE (lines) • DEVICE

N★BUS ALSO includes **BPRT,** a program print/variable cross-reference utility featuring an ultra-fast shell-Metzner sort, **BPAK,** a source program packer and **RE,** a file rename program. IF YOU USE NORTH STAR BASIC, you owe it to yourself to **ORDER YOUR COPY OF N★BUS today!**

AND ONLY $69 — plus $1.50 postage

64 pg. manual, alone, $9.75 ppd. (Creditable toward system purchase)

Calif. Res. add 6% tax. No C.O.D.'s please. Send check or M.O. VISA or MASTER CHARGE accepted.

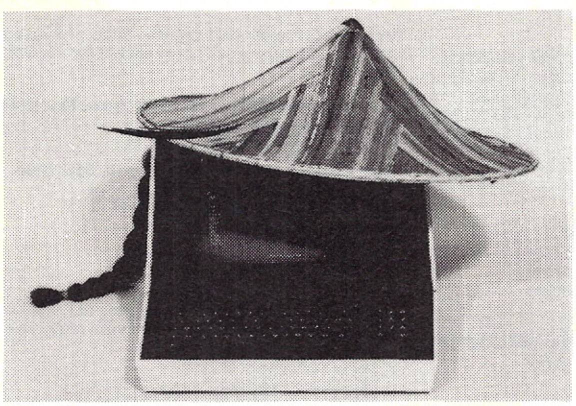
Dealer inquiries invited
COMPLETE BROCHURE AVAILABLE ✓348

SZ Software Systems

1269 Rubio Vista Rd. • Altadena, Ca. 91001

WP-6502

a very fine word processor

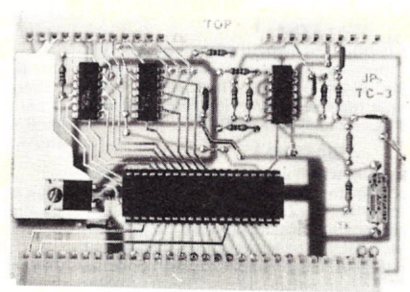


for OHIO SCIENTIFIC

Tape (C1,C2,C4) $75	8" 65D & 65U $125
5" Disk (C1,C2,C4) ...$75	Descriptive Brochure .... FREE
8" Disk for 65D $75	

Dwo Quong Fok Lok Sow
Box 4196, Grand Central Station
New York City, N.Y. 10163
(212) 685-2188 ✓87

JPC PRODUCTS FOR 6800 COMPUTERS



High Performance Cassette Interface

- **FAST** - 4800 Baud Loads 4K in 8 Seconds!
- **RELIABLE** - Error Rate Less Than 1 in 10^6 Bytes.
- **CONVENIENT** - Plugs Directly Into The **SWTPC.**
- **PLUS** - A Fully Buffered 8 Bit Output Port Provided.
- **LOW COST** - $59.95 For Complete Kit.

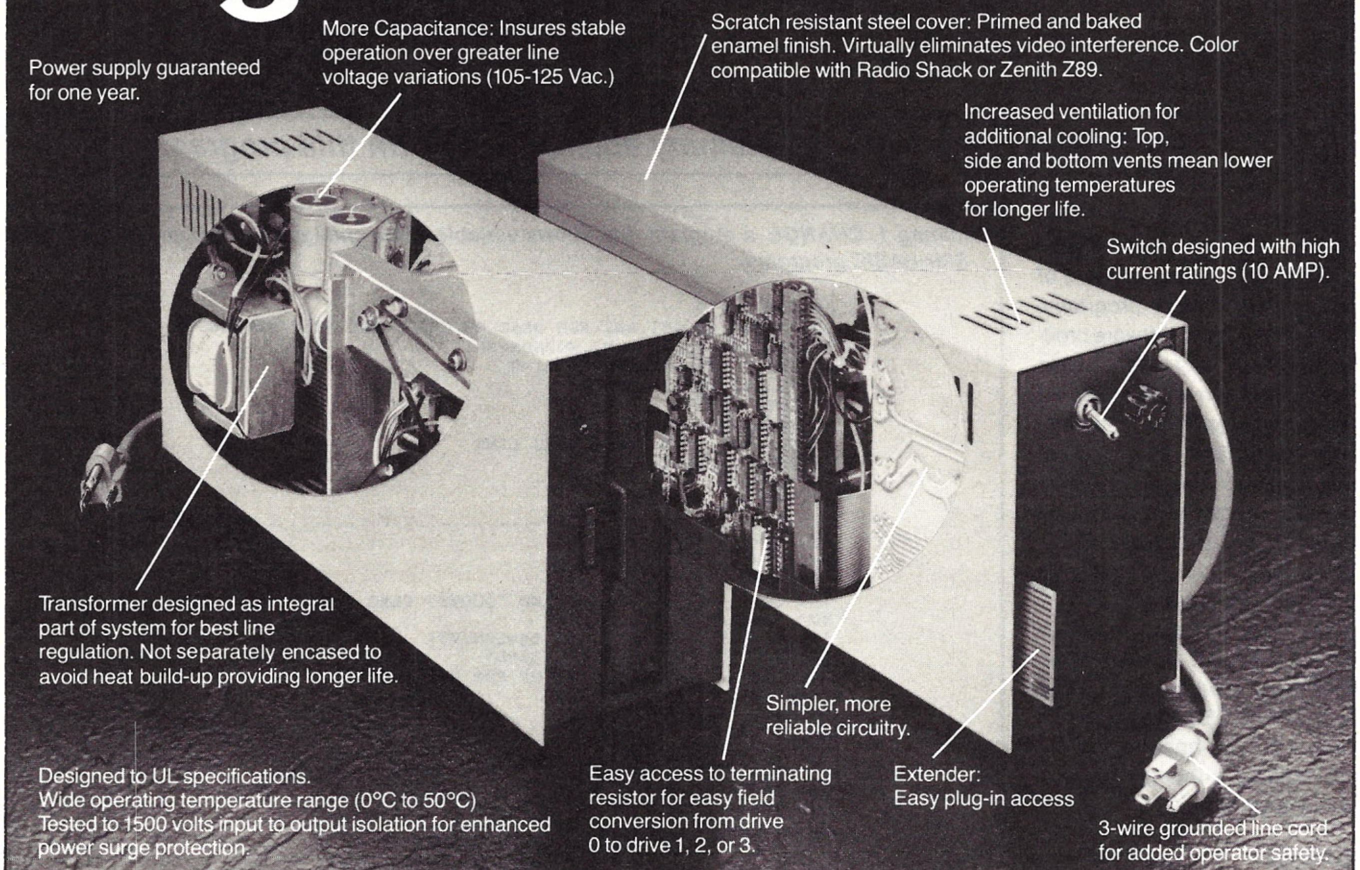
- **OPTIONAL** - CFM/3 File Manager.
  Manual & Listing $19.95
  (For Cassette Add) $ 6.95

TERMS: CASH, MC or VISA; Shipping & Handling $3.00

JPC products ✓92

Order Phone (505) 294-4623
P.O. Box 5615
Albuquerque, N.M. 87185

Precision Engineered Drives...



With the number of disk drives on the market increasing, more and more people are beginning to ask what's underneath that cover.

The CCI™ series of disk drives have been designed for long life and ease of operation. The features shown above are what set our CCI drives apart from the rest. With a CCI drive you get an integrated professional design!

If you're still not convinced that you get the most for your money with a CCI drive, just ask for our complete specifications sheet. Then, compare our disk drives to anyone else's.

5¼" DRIVES

CCI-100	40 Track (102K Bytes) for TRS-80* Model I	**$399.00**
CCI-189	40 Track (102K Bytes) for Zenith Z89	**$499.00**
CCI-200	77 Track (197K Bytes) for TRS-80* Model I	**$675.00**

8" DRIVES

CCI-800	77 Track (½ Meg Bytes) for TRS-80* Model II	**$895.00**

All CCI drives are also available for 220 Vac (50Hz) operation.

Operating Systems

NEWDOS Plus for 5¼", 40 and 77 Track Drives—with over 200 modifications and corrections to TRSDOS **$110.00**
CP/M for Model I, Zenith **$150.00**
CP/M for Model II, Altos **$250.00**

Software by S&M Systems

INSEQ-80™—Indexed Sequential Access Method (**ISAM**) for the TRS-80 Model I.

Four machine language programs that can be called from your BASIC program via USR functions to access records either sequentially or randomly. The INSEQ-80 programs maintain all indexes and chains for you. Includes reorganization utility to consolidate files. **$49.95**

Professional Business Software using INSEQ-80 for the TRS-80* Model I and Zenith Z89.

Accounts Payable, Accounts Receivable, General Ledger, Payroll per package **$99.00**
Inventory per package **$125.00**

ComputerCity™

175 Main Street, Dept. 8K, Charlestown, MA 02129 ✓278
Hours: 10AM-6PM (EST) Mon.-Fri. (Sat. till 5).

TO ORDER CALL
TOLL FREE 1-800-343-6522
TWX: 710-248-1796
Massachusetts residents call 617/242-3350
For detailed technical information, call 617/242-3350.
Freight Collect, F.O.B. Charlestown.
*TRS-80 is a trademark of the Tandy Corporation

Products also available from: Radio Shack, NEC, Centronics, Paper Tiger, TI, Altos, MPI, Zenith, Mattell, ATARI, PET, OKIDATA, Apple, Eaton/LRC.

FRANCHISE AND DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED
Retail Stores: MA: Burlington, Charlestown, Framingham, Hanover **NH:** Manchester **RI:** Providence

master charge THE INTERBANK CARD VISA

Dr. David J. Yates
Botany Department
University of Queensland
St. Lucia, 4067
Queensland, Australia

On Time and Space

North Star users can save memory and run time with this program.

The purposes of this article are to present a program I wrote to help reduce the size of North Star BASIC programs, and to describe a software product that not only saves space but reduces run times of almost any North Star BASIC program. (The North Star floating-point board allows use of a slightly smaller interpreter and, in my experience, improves *calculation* times, but for many it is an unaffordable luxury.)

One way to reduce program size is to remove all blanks from a program. A program to do this is on the NSSE Disk 2. You can save additional space by removing remarks, although this may not be desirable.

CHANGE

Two-character variable names use one more byte at each occurrence than do one-character names. I wrote the program CHANGE (Listing 1) to convert two-character to one-character variable names in North Star BASIC. It uses as data a BASIC program on file. The program requests the file name and then information about the variable names to be changed.

In Segment A, any number of variables may be altered. A two-character variable name may be altered to a single character variable plus a blank. In addition, one-character names may be changed to other one-character names.

Since North Star BASIC allows variable names consisting of only one letter and, optionally, one integer, conversion of programs from another BASIC sometimes requires the

Listing 1. CHANGE, a program that allows variable name changes and compaction in North Star BASIC programs.

```
10 GOTO  910
20 READ#0,&K,&N1,&N2\REM READ NO CHARS AND STATEMENT NO
30 WRITE#1,&K,&N1,&N2,NOENDMARK
40 LET K1=K\LET A=0\LET A1=0
50 IF K=1 THEN  830
60 LET  F=0
70 N8=N2*256+N1
80 FOR I=1 TO K-3\REM READ LINE
90 READ#0,&A9(I)
100 NEXT I
110 IF O=2 THEN  580
120 REM .........................................SEGMENT A
130 L5=0\L3=0
140 FOR J=1 TO N9
150 A1=0\A=0
160 Q=-1\R=0\REM REMARK AND "QUOTE" FLAG
170 L5=L(J)+L3
180 V$=""\V$=S$(L3+1,L5)\L4=LEN(V$)
190 W$=""\W$=T$(L3+1,L5)\L3=L5
200 FOR M=1 TOL4\REM SET UP FOR NAME LENGTH
210 V(M)=ASC(V$(M,M))
220 W(M)=ASC(W$(M,M))
230 NEXT M
240 FOR I=1 TO K-3
250 A2=A1\A1=A\REM SAVE PREVIOUS READS
260 A=A9(I)
270 I1=I
280 IF A9(I)=143 THEN R=1\REM REM
290 IF A9(I)=92 THEN R=0\REM  SLASH
300 IF A9(I)=34 AND R=0 THEN Q=-Q\REM QUOTE
310 IF Q=1 OR R=1 THEN NEXT I\REM INSIDE REM OR QUOTE
320 IF A1=154 OR A2=154 OR A1=150 THEN NEXT I\REM SKIP LINE NOS
330 IF A9(I)=38 THEN NEXT I\REM LOOK FOR BYTE ACCESS
340 FOR L=1 TO L4
350 IF A9(I)=V(L) AND Q=-1 AND R=0 THEN  400
360 IF A9(I)=92 THEN R=0\REM SLASH
370 IF A9(I)=143 THEN R=1\REM REM
380 EXIT  530
390 IF A9(I)=34 THEN Q=-Q\REM QUOTE
400 I=I+1
410 NEXTL
420 I=I-1
430 IF A9(I+1)=44 OR A9(I+1)= 41 THEN  460
440 IF A9(I+1)=13 THEN  460
450 IF I<K-4 AND A9(I+1)>32 AND A9(I+1)<91 THEN  530
460 FOR L=1 TO L4
470 A9(I1-1+L)=W(L)
480 NEXT L
490 IF F=0 THEN !
500 IF F=0 THEN !%6I,N8,TAB(15),
510 F=1
520 !W$,"     ",
530 IF I1>K-3 OR I>K-3 THEN  550
540 NEXT I
550 NEXT J
560 IF O=1 THEN  750
570 REM .........................................SEGMENT B
580 I1=0
590 Q=-1\R=0\REM REMARK AND "QUOTE" FLAG
600 FOR I=1 TO K-3
610 A2=A1\A1=A\REM SAVE PREV
620 A=A9(I)\REM GET CHARACTER
630 IF A1=154 OR A2=154 THEN  660\REM IGNORE STATEMENT NUMBERS
640 IF A =32 AND Q=-1 AND R=0 THEN  730\REM SKIP BLANK
650 IF A =128 AND Q=-1 AND R=0 THEN   730\REM SKIP LET
660 IF A<>92 THEN  680
670 Q=-1\R=0\REM NEW STATEMENT ENCOUNTERED
680 IF A=143 THEN R=1
690 IF A=34 ANDR=0 THENQ=-Q
700 I1=I1+1
710 A9(I1)=A
```

```
720 GOTO  740
730 K1=K1-1
740 NEXT I
750 FOR I=1 TO K1-3
760 WRITE#1,&A9(I),NOENDMARK
770 NEXTI
780 K9=K9+K-K1\REM NO CHARS REMOVED
790 IF K<>K1 THEN WRITE#1%Z,&K1,NOENDMARK\REM CORRECT CHAR COUNT IF NECESSARY
800 Z=Z+K1\READ#1%Z-1,&A\REM BRING POINTER UP
810 IF O=2 THEN  20
820 GOTO  20
830 !
840 IF O>1 THEN !K9,"  CHARACTERS REMOVED"
850 CLOSE#1\CLOSE#2
860 !\!
870 !"  FINISHED"
880 !\!\!
890 END
900 REM ..........................................SEGMENT C
910 !"       PROGRAM TO CHANGE VARIABLE NAMES "
920 !"  IN NORTH STAR BASIC PROGRAMS (RELEASE 4) "
930 !
940 !" WRITTEN BY      DR. D.J.YATES"
950 !"                 BOTANY DEPARTMENT"
960 !"                 UNIVERSITY OF QUEENSLAND"
970 !"                 ST. LUCIA,4067"
980 !"                 QUEENSLAND,"
990 !"                 AUSTRALIA."
1000 !\!\!\!
1010 INPUT" WHAT IS BASIC PROGRAM FILE NAME  ?",A$
1020 !
1030 LET Z=0\P=0
1040 OPEN#0%2,A$
1050 OPEN#1%2,A$
1060 !"YOU MAY:  (1)   Alter Variable Names"
1070 !"          (2)   Delete Blanks and `LET`"
1080 !"          (3)   Perform both (1) and (2)"
1090 INPUT "WHICH OPTION ? ",O
1100 IF O<1 OR O>3 THEN  1090
1110 K9=0
1120 N9=1
1130 IF O=2 THEN  1150
1140 INPUT" HOW MANY VARIABLE NAMES DO YOU WANT TO CHANGE ? ",N9
1150 N5=N9*2\REM DIMENSIONS MUST BE LARGER IF LONGER NAMES ARE REPLACED
1160 DIM A9(255),S$(N5),T$(N5),V(N5),W(N5)
1170 !
1180 IF O=2 THEN  20
1190 S$=""\T$=""
1200 FOR I=1 TO N9
1210 V$=""\W$=""
1220 !" VARIABLE NO ",I," =",\INPUT1"   ",V$
1230 L(I)=LEN(V$)
1240 S$=S$+V$
1250 INPUT"  TO BE REPLACED BY   ",W$
1260 IF LEN(W$)<=L(I) THEN  1290
1270 !" NAME MUST NOT BE LONGER THAN THAT BEING REPLACED "
1280 GOTO  1250
1290 IF LEN(W$)=L(I) THEN  1320
1300 W$=W$+" "
1310 GOTO  1290
1320 T$=T$+W$
1330 NEXT
1340 !\!
1350 IF O<>2 THEN!"STATEMENT   VARIABLE"
1360 GOTO  20
1370 END
```

conversion of two-letter names before a program is run. It is often simpler to enter a "foreign" program "as is" and then alter all occurrences of the "illegal" names at one time. This may be done with CHANGE. A danger in this program is that it will change one variable (B1, for example) to another (B) if instructed to do so. However, the interpreter will not then distinguish between the original occurrences of B and the "new" occurrences, and the program may be ruined.

By requesting the program to change to the same name all occurrences of variable name X or the foreign function LEFT or MID, CHANGE will list all occurrences of the name or function against the line numbers in which they occur, thus simplifying editing. Provided syntax is correct, none of the above changes is made within quotation marks or REM statements.

Segment B of CHANGE is based on the program by L. Steiner on NSSE 2. It allows removal of all blanks that exist after the variable name changes are made. In addition, it removes the nonessential reserved word LET from the pro-

```
10 REM PROGRAM TO PRINT THE FIRST N2 FIBONACCI NUMBERS
20 ! "FIBONACCI NUMBERS:"
30 !
40INPUT" HOW MANY NUMBERS DO YOU WANT ? ",N2
50!
60 LET N=0\REM LET INCLUDED FOR TEST OF "CHANGE"
70 LET A9=0
80 !%#10I,A9,\REM PRINT A9
90 LET N=N+1
100 B=1
110 !B,
120 N=N+1
130 C=A9+B
140 !C,
150 N=N+1
160 IF N=N2 THEN 200
170 A9=B
180 B=C
190 GOTO 130
200 !\!
210 ! N2,"  FIBONACCI NUMBERS HAVE BEEN PRINTED."
220 END
READY
```

Listing 2. FIBON, a program to print Fibonacci numbers.

```
       PROGRAM TO CHANGE VARIABLE NAMES
   IN NORTH STAR BASIC PROGRAMS (RELEASE 4)

 WRITTEN BY      DR. D.J.YATES
                 BOTANY DEPARTMENT
                 UNIVERSITY OF QUEENSLAND
                 ST. LUCIA,4067
                 QUEENSLAND,
                 AUSTRALIA.


 WHAT IS BASIC PROGRAM FILE NAME  ?FIBON

YOU MAY:  (1)   Alter Variable Names
          (2)   Delete Blanks and `LET`
          (3)   Perform both (1) and (2)
WHICH OPTION ? 3
 HOW MANY VARIABLE NAMES DO YOU WANT TO CHANGE ? 3

 VARIABLE NO  1 =   A9  TO BE REPLACED BY   A
 VARIABLE NO  2 =   N2  TO BE REPLACED BY   M
 VARIABLE NO  3 =   C  TO BE REPLACED BY   Z


STATEMENT   VARIABLE

    40         M
    70         A
    80         A
   130         A      Z
   140         Z
   160         M
   170         A
   180         Z
   210         M
 39  CHARACTERS REMOVED
```

Listing 3. Output produced by CHANGE in processing FIBON.

```
10REM PROGRAM TO PRINT THE FIRST N2 FIBONACCI NUMBERS
20!"FIBONACCI NUMBERS:"
30!
40INPUT" HOW MANY NUMBERS DO YOU WANT ? ",M
50!
60N=0\REM LET INCLUDED FOR TEST OF "CHANGE"
70A=0
80!%#10I,A,\REM PRINT A9
90N=N+1
100B=1
110!B,
120N=N+1
130Z=A+B
140!Z,
150N=N+1
160IFN=MTHEN200
170A=B
180B=Z
190GOTO130
200!\!
210!M,"  FIBONACCI NUMBERS HAVE BEEN PRINTED."
220END
READY
```

Listing 4. FIBON after being processed by CHANGE.

```
      PROGRAM TO CHANGE VARIABLE NAMES
  IN NORTH STAR BASIC PROGRAMS (RELEASE 4)

 WRITTEN BY      DR. D.J.YATES
                 BOTANY DEPARTMENT
                 UNIVERSITY OF QUEENSLAND
                 ST. LUCIA,4067
                 QUEENSLAND,
                 AUSTRALIA.


 WHAT IS BASIC PROGRAM FILE NAME  ?CHANGE

YOU MAY:  (1)    Alter Variable Names
          (2)    Delete Blanks and `LET`
          (3)    Perform both (1) and (2)
WHICH OPTION ? 3
 HOW MANY VARIABLE NAMES DO YOU WANT TO CHANGE ? 3

 VARIABLE NO  1 =   A9  TO BE REPLACED BY   B
 VARIABLE NO  2 =   K9  TO BE REPLACED BY   K9
 VARIABLE NO  3 =   F   TO BE REPLACED BY   G


STATEMENT   VARIABLE

    60          G
    90          B
   260          B
   280          B
   290          B
   300          B
   330          B
   350          B
   360          B
   370          B
   390          B
   430          B       B
   440          B
   450          B       B
   470          B
   490          G
   500          G
   510          G
   620          B
   710          B
   760          B
   780          K9      K9
   840          K9
  1110          K9
  1160          B
 351   CHARACTERS REMOVED
```

Listing 5. Output produced by CHANGE in processing itself.

gram. At the end of processing the program statements, the abbreviated program is written back to the original file. Minor modification allows the output file to be different from the input file. It is also easy to delete the optional GOTO in statements such as IF . . . THEN GOTO 123.

Example

Listing 2 is a program, FIBON, in its "raw" state before being processed by CHANGE. Listing 3 represents the output produced by CHANGE when FIBON is processed. Listing 4 is FIBON after being processed by CHANGE, with the altered names referred to in Listing 3. Note that names N2 and A9 in lines 10 and 80, respectively, are not changed. As indicated in Listing 3, blanks and LETs are deleted to save 39 bytes, about 11 percent of the total for the original program, which did not have excessive blanks, LETs or two-character names. Listing 5 indicates the method of finding all occurrences of the variables K9 and F in CHANGE itself, as well as changing B for A9. You can save a significant number of bytes by using the techniques described. Listing 6 represents a portion of CHANGE after being processed by itself.

```
490IFG=0THEN!
500IFG=0THEN!%6I,N8,TAB(15),
510G=1
520!W$,"     ",
530IFI1>K-3ORI>K-3THEN550
540NEXTI
550NEXTJ
560IFO=1THEN750
570REM .........................................SEGMENT B
580I1=0
590Q=-1\R=0\REM REMARK AND "QUOTE" FLAG
600FORI=1TOK-3
610A2=A1\A1=A\REM SAVE PREV
620A=B(I)\REM GET CHARACTER
630IFA1=154ORA2=154THEN660\REM IGNORE STATEMENT NUMBERS
640IFA=32ANDQ=-1ANDR=0THEN730\REM SKIP BLANK
650IFA=128ANDQ=-1ANDR=0THEN730\REM SKIP LET
660IFA<>92THEN680
670Q=-1\R=0\REM NEW STATEMENT ENCOUNTERED
680IFA=143THENR=1
690IFA=34ANDR=0THENQ=-Q
700I1=I1+1
710B(I1)=A
720GOTO740
730K1=K1-1
740NEXTI
750FORI=1TOK1-3
760WRITE#1,&B(I),NOENDMARK
770NEXTI
780K9=K9+K-K1\REM NO CHARS REMOVED
790IFK<>K1THENWRITE#1%Z,&K1,NOENDMARK\REM CORRECT CHAR COUNT IF NECESSARY
800Z=Z+K1\READ#1%Z-1,&A\REM BRING POINTER UP
810IFO=2THEN20
820GOTO20
830!
840IFO>1THEN!K9,"  CHARACTERS REMOVED"
850CLOSE#1\CLOSE#2
READY
```

Listing 6. Portion of CHANGE after processing by itself.

Changes and Deletions

On file, and in memory, a North Star BASIC statement is stored as a string of hexadecimal numbers. The first is a character count (N) between 0 and 255. The second two numbers carry the line number. The remaining N – 3 numbers are the code representing the rest of the statement. The reserved words are not stored

!	146	DIM	139	OPEN	151
(	224	ELSE	180	OR	237
+	227	END	141	OUT	148
-	229	ERRSET	159	PRINT	130
/	231	EXAM	218	PSIZE	174
<	244	EXIT	150	READ	133
<=	240	EXP	222	REM	143
<>=	241	FILL	149	REN	168
=	245	FN	144	RESTORE	142
>	246	FOR	129	RETURN	138
>=	239	GOSUB	137	RND	206
ABS	219	GOTO	136	RUN	160
AND	236	IF	132	SAVE	170
ASC	182	INP	217	SCR	163
AUTO	164	INPUT	134	SIN	203
CALL	205	LEN	204	SQRT	196
CAT	175	LET	128	STEP	176
CHAIN	155	LINE	156	STOP	140
CHR$	181	LIST	161	STR$	184
CLOSE	152	LOAD	165	TAB	179
CONT	166	LOG	221	THEN	178
COS	220	MEMSET	162	TO	177
CREATE	158	NEXT	131	TYP	223
DATA	135	NOENDMARK	185	VAL	183
DEF	145	NOT	247	WRITE	153
DEL	173	NSAVE	169	\	154
DESTROY	157	ON	147	^	225

Table 1. North Star BASIC reserved words and their decimal token requirements.

as their full ASCII representations, but in an abbreviated form in which each is represented by one hexadecimal number. Most of the other alphanumeric characters signifying operators that BASIC recognizes are also stored in a coded form—not their ASCII values but values greater than 127 (See Table 1).

This saves space required to change the program. CHANGE searches for the desired first character in the variable name using its ASCII code. On finding the first character it continues to look for the rest of the name and makes the change when it finds it. If the variable in the original line is longer than the one specified, no change is made. On encountering a quotation mark or REM, CHANGE attempts no changes until after the next quotation or the end of the REM.

In the blank and LET detection routine, when a character is detected in the search, it is deleted, all "characters" are moved along one space, and the character count is reduced by one. The magnitude of line numbers per se does not influence execution time because two bytes represent the line number for all lines.

Save Time, Too

I bought a $29 (U.S.) software product called DOC, a valuable utility package. It is marketed by

```
10 FOR J=1 TO 10000
20 GOTO 30
30 NEXT
40 !"STOP"\END
50REM
60REM
70REM
80REM
90REM
100REM
110REM
120REM
130REM
140REM
150REM
160REM
170REM
180REM
190REM
200REM
210REM
220REM
230REM
240REM
250REM
260REM
270REM
280REM
290REM
300REM
310REM
320REM
330REM
340REM
350REM
360REM
370REM
380REM
390REM
400REM
410REM
420REM
430REM
440REM
450REM
460REM
470REM
480REM
490REM
500 GOTO 30
```

Listing 7. Program showing processing of GOTOs.

✓Reader Service index—page 241

'CHANGE' - VARIABLE CHANGE VARIABLE MAP

VARIABLE	USED IN LINE						
A	40	150	250	260	610	620	640
	650	660	680	690	710	800	
A$	1010	1040	1050				
A1	40	150	250	250	320	320	610
	610	630					
A2	250	320	610	630			
A9	90	260	280	290	300	330	350
	360	370	390	430	430	440	450
	450	470	620	710	760	1160	
F	60	490	500	510			
I	80	90	100	240	260	270	280
	290	300	310	320	330	330	350
	360	370	390	400	400	420	420
	430	430	440	450	450	450	500
	530	540	600	620	740	750	760
	770	1200	1220	1230	1260	1290	
I1	270	470	530	580	700	700	710
J	140	170	550				
K	20	30	40	50	80	240	450
	530	530	600	780	790		
K1	40	730	730	750	780	790	790
	800						
K9	780	780	840	1110			
L	170	340	350	410	460	470	470
	480	1230	1260	1290			
L3	130	170	180	190	190		
L4	180	200	340	460			
L5	130	170	180	190	190		
M	200	210	210	210	220	220	220
	230						
N1	20	30	70				
N2	20	30	70				
N5	1150	1160	1160	1160	1160		
N8	70	500					
N9	140	1120	1140	1150	1200		
O	110	560	810	840	1090	1100	1100
	1130	1180	1350				
P	1030						
Q	160	300	300	310	350	390	390
	590	640	650	670	690	690	
R	160	280	290	300	310	350	360
	370	590	640	650	670	680	690
S$	180	1160	1190	1240	1240		
T$	190	1160	1190	1320	1320		
V	210	350	1160				
V$	180	180	180	210	1210	1220	1230
	1240						
W	220	470	1160				
W$	190	190	220	520	1210	1250	1260
	1290	1300	1300	1320			
Z	790	800	800	800	1030		

Listing 8. The variable map produced by DOC in processing CHANGE.

Mini Business Systems (PO Box 15587, Salt Lake City, UT 84115) in BASIC. DOC allows you to list a BASIC program; list all variables in a BASIC program and the number of each line in which each appears; list GOTOs and GOSUBs on a "from-to" basis; and optimize programs.

Listing is done on a page basis. Page length, spacing and output device are selectable, and a title, date and page number are placed at the top of each page. The listings of the variables and GOTO/GOSUBs are useful, particularly in debugging large programs, but really are by-products of the preparation for optimization.

The optimization is a gem. In executing GOTO/GOSUBs, the North Star interpreter "starts at the top" and looks at successive line numbers until it finds the right one. The time required to execute a GOTO depends on how many lines down the program the specified line lies. If you run the program in Listing 7 and measure execution time, then repeat the process with line 20 changed to GOTO 500, you will see the significance of the search technique employed.

I ran Listing 6 in 13.9 seconds . . . 48 seconds when I altered line 20 to GOTO 500. The difference between the two runs was only the number of statement numbers scanned. Successively removing REMs from this program shows that the interpreter takes approximately 66 microseconds to scan a statement number. This mounts up with GOTOs and GOSUBs within loops. Anything that places the subroutines and line numbers most frequently "gone to" near the top of the program speeds up the program.

A reduction in the number of lines in a program also reduces the amount of searching required. Concatenating statements reduces the number of lines, and DOC uses this technique effectively.

In concatenating, you can remove REMs to optimize the program even further. Surprisingly, LET is not removed. DOC programs actually print the optimized program if required. Optimized programs are often barely readable and, if longer than the line-length maximum (132 characters) allowed by North Star, may not be edited or listed. Optimized programs are often much faster than their non-optimized starting point. The original copy is preserved if required.

I used DOC to optimize CHANGE. Listing 8 and 9 show portions of the output produced by DOC, and Listing 10 is what the optimized program would look like if it could be listed by the interpreter. The original program was 3274 bytes, the optimized version is 2010 bytes, a 39 percent reduction in memory requirement. When run again, the optimized CHANGE performed the same function referred to in Listings 3 and 4. Total run time was 11 minutes, 22 seconds, compared with the original run time of 13 minutes, 45 seconds, a saving of 17 percent in time.

I have three criticisms of the

'CHANGE' - VARIABLE CHANGE

GOTO	FROM			
20	810	820	1180	1360
400	350			
460	430	440		
530	380	450		
550	530			
580	110			
660	630			
680	660			
730	640	650		
740	720			
750	560			
830	50			
910	10			
1090	1100			
1150	1130			
1250	1280			
1290	1260	1310		
1320	1290			

Listing 9. GOTO map produced by DOC in processing CHANGE.

```
10GOTO 910
 20READ#0,&K,&N1,&N2\WRITE#1,&K,&N1,&N2,NOENDMARK\LETK1=K\LETA=0\LETA1=0\IFK=1THEN 830\LETF=0\N8=N2*256+N1\FORI=1TOK-3\READ#0,&A9(I)
\NEXTI\IFO=2THEN 580\L5=0\L3=0\FORJ=1TON9\A1=0\A=0\Q=-1\R=0\L5=L(J)+L3\V$=""\V$=S$(L3+1,L5)\L4=LEN(V$)\W$=""\W$=T$(L3+1,L5)\L3=L5\FO
RM=1TOL4\V(M)=ASC(V$(M,M))
 220W(M)=ASC(W$(M,M))\NEXTM\FORI=1TOK-3\A2=A1\A1=A\A=A9(I)\I1=I\IFA9(I)=143THENR=1\IFA9(I)=92THENR=0\IFA9(I)=34ANDR=0THENQ=-Q\IFQ=1O
RR=1THENNEXTI\IFA1=154ORA2=154ORA1=150THENNEXTI\IFA9(I)=38THENNEXTI\FORL=1TOL4\IFA9(I)=V(L)ANDQ=-1ANDR=0THEN 400\IFA9(I)=92THENR=0\I
FA9(I)=143THENR=1\EXIT 530\IFA9(I)=34THENQ=-Q
 400I=I+1\NEXTL\I=I-1\IFA9(I+1)=44ORA9(I+1)=41THEN 460\IFA9(I+1)=13THEN 460\IFI<K-4ANDA9(I+1)>32ANDA9(I+1)<91THEN 530
 460FORL=1TOL4\A9(I1-1+L)=W(L)\NEXTL\IFF=0THEN!\IFF=0THEN!%6I,N8,TAB(15),\F=1\!W$,"     ",
 530IFI1>K-3ORI>K-3THEN 550\NEXTI
 550NEXTJ\IFO=1THEN 750
 580I1=0\Q=-1\R=0\FORI=1TOK-3\A2=A1\A1=A\A=A9(I)\IFA1=154ORA2=154THEN 660\IFA=32ANDQ=-1ANDR=0THEN 730\IFA=128ANDQ=-1ANDR=0THEN 730
 660IFA<>92THEN 680\Q=-1\R=0
 680IFA=143THENR=1\IFA=34ANDR=0THENQ=-Q\I1=I1+1\A9(I1)=A\GOTO 740
 730K1=K1-1
 740NEXTI
 750FORI=1TOK1-3\WRITE#1,&A9(I),NOENDMARK\NEXTI\K9=K9+K-K1\IFK<>K1THENWRITE#1%Z,&K1,NOENDMARK\Z=Z+K1\READ#1%Z-1,&A\IFO=2THEN 20\GOTO
 20
 830!\IFO>1THEN!K9,"  CHARACTERS REMOVED"\CLOSE#1\CLOSE#2\!\!\!"  FINISHED"\!\!\!\END
 910!"       PROGRAM TO CHANGE VARIABLE NAMES "\!"  IN NORTH STAR BASIC PROGRAMS (RELEASE 4) "\!\!" WRITTEN BY     DR. D.J.YATES"\!"
            BOTANY DEPARTMENT"\!"                    UNIVERSITY OF QUEENSLAND"\!"                    ST. LUCIA,4067"
 980!"                 QUEENSLAND,"\!"                 AUSTRALIA."\!\!\!\!\INPUT" WHAT IS BASIC PROGRAM FILE NAME  ?",A$\!\LETZ=0\P=0\
OPEN#0%2,A$\OPEN#1%2,A$\!"YOU MAY:  (1)   Alter Variable Names"\!"              (2)   Delete Blanks and `LET`"
 1080!"          (3)   Perform both (1) and (2)"
 1090INPUT"WHICH OPTION ? ",O\IFO<1ORO>3THEN 1090\K9=0\N9=1\IFO=2THEN 1150\INPUT" HOW MANY VARIABLE NAMES DO YOU WANT TO CHANGE ? ",
N9
 1150N5=N9*2\DIMA9(255),S$(N5),T$(N5),V(N5),W(N5)\!\IFO=2THEN 20\S$=""\T$=""\FORI=1TON9\V$=""\W$=""\!" VARIABLE NO ",I," =",\INPUT1"
  ",V$\L(I)=LEN(V$)\S$=S$+V$
 1250INPUT"  TO BE REPLACED BY   ",W$\IFLEN(W$)<=L(I)THEN 1290\!" NAME MUST NOT BE LONGER THAN THAT BEING REPLACED "\GOTO 1250
 1290IFLEN(W$)=L(I)THEN 1320\W$=W$+" "\GOTO 1290
 1320T$=T$+W$\NEXT\!\!\IFO<>2THEN!"STATEMENT   VARIABLE"\GOTO 20\END
```

Listing 10. CHANGE after processing by DOC.

copy of DOC I received. The reserved word ERRSET was not included in its "repertoire"—not really a problem since Mini Business Systems has devised a clever way of updating DOC to cater to all new reserved words in future releases of North Star BASIC. Second, BASIC programs have to be converted to data files before being processed by DOC. This is annoying, particularly because the Release 4 of North Star BASIC used allows any file type to be accessed as a data file. The most glaring problem with the system is that it doesn't handle multiline functions properly. A program containing functions will not run if it has been optimized, but you can use a feature of DOC to prevent optimization of segments within a program. This overcomes the problem.

Overall, the DOC package is excellent. A nice bonus is a small program called GOTOSUB, a clever routine that is an implementation of the "GOTO N" statement. It is used in part of the DOC package and could significantly improve the runtime of many BASIC programs. ■

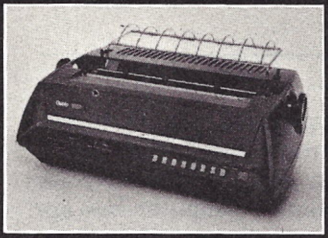
Data Terminals From MICROMAIL? YES,

Because We Offer....

...A 'Personal Approach'
Towards the Quick and Efficient Handling of Your Individual Order.

...Inventory.
The Equipment You Select is Readily Available from Our Stock.

...Terminals Only.
We Specialize in Data Terminal Equipment.



DIABLO 1650

- Prints at 40 cps, using 88, 92, or 96 char. metalized printwheels.
- Vertical resolution 1/48"; Horizontal 1/120", Capable of proportional spacing, bidirectional printing, and graphics under software control.
- Bidirectional normal and direct tabs. Left, right, top and bottom margins.

R.O. $2890.00
KSR $3285.00

DIABLO 1640

- Uses plastic printwheel and prints at 45 cps. Otherwise, shares identical features with 1650 including:
  - — Friction or tractor feed, up to 15" wide.
  - — Cartridge ribbon, fabric or carbon.

R.O. $2745.00
KSR $3140.00

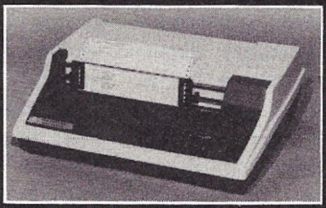
DEC LA 34

(Shown with optional forms tractor and numeric keypad).

- Prints 10, 12, 13.2, or 16.5 characters per inch, upper/lower case.
- 2, 3, 4, 6, 8, or 12 lines per inch.
- Friction feed, paper width to 15 inches.

$999.00

Options:
— Numeric keypad — **$80.00**
— Adjustable forms tractor — **$130.00**



T.I. 810

- Includes upper/lower case option.
- Bidirectional printing at 150 cps.
- Tractor-feed forms, 3" to 15" wide.

$1599.00

Options:
— Forms length control — **$90.00**
— Vertical Format Control with Compressed Print — **$253.00**

SOROC IQ 120

- Displays 80 x 24, upper/lower case.
- Separate numeric keypad and cursor keys.
- Protected fields displayed at reduced intensity.

$740.00

SOROC IQ 140

- 117-key detachable keyboard with numeric cluster and cursor control.
- Insert/delete line, insert/delete character.
- Underline, blink, reverse, 1/2 intensity, protected and blank fields.
- Printer port with independent baud rate — prints line, partial or full screen.

$1130.00

TEC 510

- Reverse video, blinking, underline, 1/2 intensity, protected fields, blank security field.
- Transmit character, line, partial page, page, or unprotected data.
- Cursor up, down, left, right, return, home, plus load and read.

$750.00
(in quantities)

TELETYPE 43

- Prints 132 columns, upper/lower case with true descenders.
- 30 character/second print speed. 110-300 baud.
- Uses 12" wide by 8.5" pinfeed paper.
- Print position scale, paper guide and supply rack.

$999.00

To Order:
Send check or money order to: **MICROMAIL, P.O. Box 3297, Santa Ana, CA 92703.** Personal or company checks require two weeks to clear.

Terminals in stock are shipped the business day after receipt of certified funds.

All equipment includes factory warranty.

MICROMAIL™

MICROMAIL • BOX 3297 • SANTA ANA, CA 92703
(714) 731-4338

68

Shipping:
We ship freight collect by UPS when possible. Larger terminals are shipped by motor freight. Air and express delivery is available on all products.

Why Do You Need Two Disks?

A good question. Here are some answers.

James W. Stutsman
3139 Oak Hill Rd.
Carrollton, TX 75007

In the early days of computers (before Intel discovered silicon), there were no disk operating systems, or disks, for that matter. Computers of the day were monstrous and used large amounts of power and took up massive amounts of space. Some of the larger ones had nearly the power of an 8008.

The programmers of this time had to do everything on tape or punched cards. Any required operating system functions were included with their programs. The high cost of computing made this scheme impractical, although these computers were a type of personal computer.

In an attempt to recover some of the time lost when the computer changed from one user to another, the batch processing operating system was born. Users submitted their programs and data on punched cards, which were read onto a tape. When the tape became full, it was processed and the output was distributed to the users.

Improvements in hardware architecture, including the introduction of random access disk drives, allowed operating systems to become more versatile. This, in turn, allowed more effective use of the available computer power. These operating systems relied upon the disk as their primary storage device and thus became generally known as disk operating systems (DOS). Even now, most of the operating systems used on the larger computers are disk-based. However, all of this convenience and power consumes as much as 40 percent of the computer's resources.

One Small Step

With the introduction of personal computers in 1975, the industry changed irrevocably. Now the power of the computer could be allocated in a one-on-one basis economically. Evolution of the operating systems for personal computers followed that of their larger brethren. These first computers had no operating system, and the programmer had to do everything (in between RAM failures). Mercifully, the batch processing stage was bypassed, probably because keypunch machines were beyond hobbyist budgets.

When floppy disks for microcomputers were introduced, the first DOS programs began to appear. These have been written mainly by big-computer programmers determined to bring the world a better DOS. Some of these efforts have been quite remarkable, producing big-computer operating systems in miniature. This has led people to rally behind their favorite DOS, forsaking all others, and attempting to convert nonbelievers.

Shortcomings

Unfortunately, it has become the norm to evaluate new computers and disk systems by the power of their DOS. Too much emphasis has been placed on the importance of DOS. Sophisticated disk operating systems are not necessary on personal computers and are actually detrimental to them. The current crop of disk operating systems has been written by programmers for programmers to use. Users of such systems may have to read a 60-page manual (or worse, have no manual) just to use them. This may, in part, explain why the so-called "home" computer has appeared only in the homes of computer professionals.

Secondly, these systems solve problems that don't exist. One example is in the area of file allocation. Most of the disk operating systems now offered have complex logic in them to allocate disk space to a program dynamically as it runs. This means that every sector of every disk will be used. The cost for this service, however, is a general slowing of disk read/write operations and making random access of disk files difficult to impossible. All of this to conserve real estate on a diskette that costs $4.50?

Disk operating systems are programs that make demands on the resources of a computer. Typical DOS memory requirements range from 8K to as much as 20K bytes of memory. This means that the user either has to restrict his memory usage or buy more memory to use a disk. Once he has his disk, he will find that the DOS wants a piece of it, too. Thirty percent or more of the system disk is used for DOS-related data, making it unavailable to the user. Thus, two drives may have to be used even though all of the data would fit on one.

A typical DOS may consist of several thousand lines of source code. In a program of this size even the best, most experienced, programmer may make a mistake. Large-computer manufacturers expect and plan for software crashes. They have support personnel available to assist the user in the recovery of his data files. A

personal computer buyer has no way of knowing about this possibility and nowhere to turn when it occurs.

DOS Requirements

What constitutes a good DOS for a personal computer? Several key items need to be provided:

1. It must provide for a simple means of saving and loading programs.

2. It must provide the necessary basics to access the disk in a direct (i.e., random) manner from a program written in a high-level language such as BASIC. All input and output functions should be handled by the DOS, as well as head positioning, error retries and error recovery.

3. It must make minimal demands on RAM memory, preferably using less than 8K. No part of the disk should be used except for the minimal amount needed to store the DOS itself.

DOS owners may wonder, "How can you do anything with such a simple DOS? You haven't even provided for a directory!" The whole point is that the application program will provide whatever DOS features it needs without bearing the overhead of a lot of features that it does not need.

For example, if a program uses all of a disk for a single inventory file, does it really need a directory of that disk? If a directory is needed, why not build it in the form most useful to the application? In this way, the least demands are made on the resources of the computer while providing the programmer with the capability to mold the operating system to best suit his application.

If the programmer can access the entire disk, then he has to do his own file allocation. He may even overwrite a file destroying the one following it! To remedy this the programmer has to plan how his files will grow and build that knowledge into the application. Isn't it better to take care of that at the time the program is written than to build a timebomb that expires with "DISK FULL IN 600"?

At first glance, it may appear that all you really have with your simple DOS is a fast cassette. Any feature of a big DOS that a user needs can be implemented in high-level language by the user. He also has full random access capability of the entire diskette. Look at a complex DOS whose threaded files prevent random access and tell me again who has the "fast cassette."

It is not my intention to downgrade any of the disk operating systems being marketed today for the personal computer hobbyist. If a big DOS is your main interest, then by all means use it. But don't make the assumption that such a DOS is essential for successfully installing an application on a personal computer. ■

> **It may appear that all you really have with your simple DOS is a fast cassette.**

NOW *Fastest Data Manager* **PET™**

JINSAM™

- ★ CUSTOM DATA FILES
- ★ CUSTOM REPORTS/LABELS
- ★ KEYED RANDOM ACCESS
- ★ FAST/EASY/MENU DRIVEN
- ★ MULTIPLE SEARCH KEYS
- ★ PRIVACY ACCESS CODES
- ★ WILD CARD SEARCH

Specify CBM 2040 or COMPU/THINK DISK

INTRODUCTORY OFFER
PACKAGE **$150**

USERS GUIDE ONLY $25
DEMO TAPE $5 DISK $8
(REFUNDABLE WITH PURCHASE)

Powerful user commands. Self explanatory, easy to use. Straight forward input and editing routines – "idiot proof". Create any desired relationship.

Optional STATISTICS, MATH, PTE WORD PROCESSOR packages

TYPICAL APPLICATIONS: Personnel files, Customer files, Inventory, Sales records, School records, Appointment schedules, Real estate /Apartment listings, Subscription lists, Research surveys, Mailings.

Send Check or Money Order plus $2 Shipping
(NY residents add 8% Sales Tax)

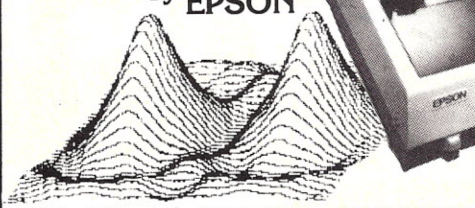
- DEALER INQUIRIES WELCOMED -

JINI MICRO-SYSTEMS, Inc. ✓164
P.O. Box 274-K • Bronx, NY 10463

NOW PRINT APPLE®

HIRES-GRAPHICS NORMAL, INVERSE, IN 2 SIZES.

TX-80 PRINTER
by EPSON



THE TX-80 MATRIX PRINTER

$795. with GRAFTRAX

SPEED: 58 LPM, 125 CPS

INTERFACES: Parallel standard IEEE488 and serial RS-232 optional, (Apple type parallel card and cable .... **$99.**)

CHARACTER SET: Full 96 Character ASCII Set (upper and lower case with expanded print).

PRINT HEAD: 100 x 10^6 character life expectancy.

GRAFTRAX OPTION* full dot addressable graphics (480 dots/line) with Automatic print head protection on dense pictures plus form feed and skip over perforation.

FREE! **APPLESOFT-WARE for graphics dump included**

*UPDATE EARLIER TX-80's TO GRAPHICS for **$99.**

Masterchrage & Visa O.K.
DEALER INQUIRIES INVITED ✓97

Computer Corner of New Jersey
439 Rt. 23, Pompton Plains, N.J. 07444 (201) 835-7080
PRICES SUBJECT TO CHANGE

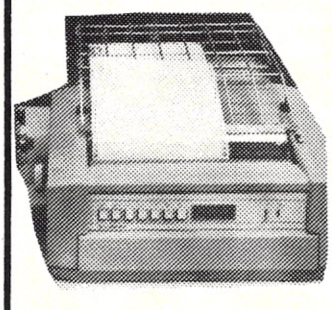
RONDURE COMPANY

✓74

2522 BUTLER ST.
DALLAS, TEXAS 75235
214–630-4621

the computer room

SPECIAL
Printer for your Microcomputer



GE TERMINET 300 PRINTER

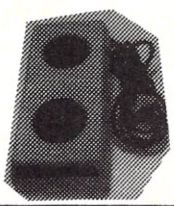
Pin feed—9" paper

- 80 Print positions
- Receive only
- ASCII code
- RS-232 interface
- 30 CPS
- Upper & lowercase
- Shipping wt. 75#

Shipping containers $15.00.
(used)
(good working condition)

Will run on serial RS232 port of most micros including TRS-80. **$450.00**

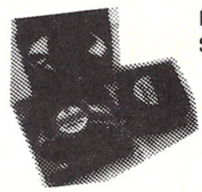
WE HAVE FLAT-PACK ACOUSTIC



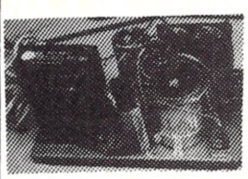
Modem pickup

$19^{50}

USED FANS



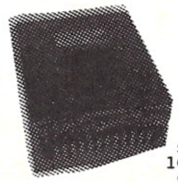
Muffin – 8.00
Sprite – 4.00



NEW POWER SUPPLY

$25.00

5V at 3 Amp
12V at 6 Amp
-12V at 3 Amp



USED POWER SUPPLY

$15.00

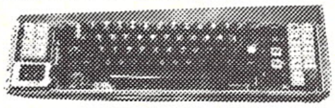
5V at 12 Amp
16V at 6 Amp
6V at 2 Amp

NEW POWER SUPPLY

(AC-DC Brand)

Model 1-22V @ 1.9a $20.00
Model 2-15V @ 2.4a $30.00

MICRO SWITCH KEYBOARD
USED BUT LOOKS VERY NICE



ASC II **$40.00**
(With Print)

USED OMNITEK

MODEM
ORIGINATE
ONLY
TESTED

$90.00 Sale

ORDERING INFORMATION:
We ship the same day we receive a certified check or money order. Texas residents add 5% sales tax. Please call if you have a question. Write for our CATALOG of many parts, terminals, printers, etc. All items subject to availability. Your money returned if we are out of stock.

SHIPPING INFORMATION:
Modems: $3.00 each; Key Boards $4.00, Power Supply $7.00.
Large Items & Parts: Specify Freight or Air Freight Collect.
Foreign Orders: Add appropriate freight or postage.
We now take Master Charge and Visa orders. Specify full number, bank number and expiration date.

✓ Reader Service index—page 241

Disassembler For the Heath H8

The DIS-8 program is self contained and should be adaptable to other 8080 and 8085 systems.

Patrick Swayne
290 Springdale
Sebastopol, CA 95472

Adapting software from one computer make to another can be a challenge, especially if the software needs RAM where there isn't any and no source listing is available. This is the problem I encountered when I decided to adapt Cromemco's Control BASIC to my Heath H8.

Control BASIC is designed to reside in ROM starting at 344000A (A is Heath's designation for split octal notation) and use RAM starting at 2000A. There is no problem with the high address, but Heath requires the first 8K + of RAM in their machine, and user RAM starts at 040100A.

The obvious need to make some changes to Control BASIC prompted me to write the disassembler (see Listing 1). This program (DIS-8) occupies about 2.5K of memory space at the bottom of the H8's user RAM area. It is self sufficient, including I/O routines, and could easily be adapted to other 8080- or 8085-based computers. (Note: The console driver was written for the H8-5 interface card. The newer H8-4 card requires different software.)

Program Operation

The output of DIS-8 resembles that of Heath's assembler, HASL-8. Each line contains the current address in split octal, the op code that is there, the data that follows two- and three-byte instructions, the mnemonic for the instruction, with data, if any, and a comment based on the ASCII value of the op code and data. The comments, which are always introduced by an asterisk (*), are produced as follows.

First, the most significant bit is stripped, and the result is checked. If it is a lowercase letter (over 137Q), it will be capitalized. This allows operation on Heath's H9 terminal or other uppercase-only terminals. If the number is below 137Q, it will be checked to see if it is below 041Q, in which case it will not be printed, and a space will be left in its place. A sample run of DIS-8 is shown in Listing 2.

In order to produce neat columns with plenty of space between them, an output buffer is used. Each part of a line of disassembly is placed into its position in the buffer; then the entire buffer is printed and cleared for the next line by filling it with ASCII spaces. To ensure the correct number of spaces between the parts of two-part mnemonics (such as "STAX   B"), a number, included in the table of mnemonics, indicates how many spaces to use. Thus, "STAX   B" appears in the table as 'STAX',6,'B'.

Disassembly is accomplished by first fetching the op code pointed to by the address pointer. The instruction is then compared to a table of two- and three-byte op codes, and the program then branches to one of three routines. In the case of one- and two-byte instructions, the op code and data, if any, are placed in the buffer. Then the mnemonic printing routine is called, which looks up the op

```
ENTER STARTING ADDRESS:  040 100

040.100  006 172          MVI       B,172Q         * Z
040.102  303 107 040      JMP       040107A        *CG
040.105  006 316          MVI       B,316Q         * N
040.107  076 201          MVI       A,201Q         *>
040.111  323 373          OUT       373Q           *SC
040.113  076 100          MVI       A,100Q         *>@
040.115  323 373          OUT       373Q           *SC
040.117  177              MOV       A,A            *_
040.120  170              MOV       A,B            *X
040.121  323 373          OUT       373Q           *SC

ENTER STARTING ADDRESS:  042 074

042.074  104              MOV       B,H            *D
042.075  111              MOV       C,C            *I
042.076  123              MOV       D,E            *S
042.077  055              DCR       L              *-
042.100  070              DB        070Q           *8
042.101  054              INR       L              *,
042.102  040              DB        040Q           *
042.103  126              MOV       D,M            *V
042.104  105              MOV       B,L            *E
042.105  122              MOV       D,D            *R
```

Listing 2. Sample run of DIS-8 disassembling parts of itself. Note program's handling of string data. (The byte loaded into the B register in the first line had to be changed to meet the requirements of the terminal.)

```
ENTER STARTING ADDRESS:  2040

2040  06 7A          MVI       B,7AH       ; Z
2042  C3 47 20       JMP       2047H       ;CG
2045  06 CE          MVI       B,CEH       ; N
2047  3E 81          MVI       A,81H       ;>
2049  D3 FB          OUT       FBH         ;SC
204B  3E 40          MVI       A,40H       ;>@
204D  D3 FB          OUT       FBH         ;SC
204F  7F             MOV       A,A         ;_
2050  78             MOV       A,B         ;X
2051  D3 FB          OUT       FBH         ;SC

ENTER STARTING ADDRESS:  2229

2229  44             MOV       B,H         ;D
222A  49             MOV       C,C         ;I
222B  53             MOV       D,E         ;S
222C  2D             DCR       L           ;-
222D  38             DB        38H         ;8
222E  2C             INR       L           ;,
222F  20             DB        20H         ;
2230  56             MOV       D,M         ;V
2231  45             MOV       B,L         ;E
2232  52             MOV       D,D         ;R
```

Listing 3. DIS-8 sample run (hex version).

```
*
*         ASCII TO BINARY CONVERSION ROUTINE
*
BIN       SUI       30H        NO CHAR < '0'
          RC
          ADI       30H-47H    NO CHAR > 'F'
          RC
          ADI       6          A THRU F?
          JP        BINO       YES, BRANCH
          ADI       7          ':' TO '@' ILLEGAL
          RC
BINO      ADI       10         ADJUST
          ORA       A          CLEAR FLAGS
          RET
--

*
*         GET STARTING ADDRESS
*
ADDR      LXI       H,0        CLEAR H,L
ADDRO     CALL      IN         GET ENTRY
          CALL      OUT        ECHO
          CPI       ODH        DONE?
          RZ                   IF SO, RETURN
          CALL      BIN        CONVERT TO BINARY
          JC        ERR        BAD ENTRY
          DAD       H          MOVE LAST ENTRY
          DAD       H          OVER 4 PLACES
          DAD       H
          DAD       H
          ORA       L          ADD LATEST ENTRY
          MOV       L,A        TO L REG
          JMP       ADDRO      GET ANOTHER ENTRY
--
```

Listing 4. Hex I/O routines (BIN, ADDR, PADDR, ASC).

```
*
*         PRINT ADDRESS
*
PADDR     MOV       A,H        GET ADDR HI BYTE
          LXI       D,BUFF
          CALL      ASC        PRINT IT
          MOV       A,L        GET LOW BYTE
*         JMP       ASC        PRINT IT
--

*
*         BINARY TO ASCII CONVERSION ROUTINE
*
ASC       PUSH      PSW        SAVE BYTE
          RRC                  MOVE HIGH NIBBLE
          RRC                  DOWN
          RRC
          RRC
          ANI       OFH        STRIP LOW NIBBLE
          CALL      ASCO       CONVERT TO ASCII
          POP       PSW        RESTORE BYTE
          ANI       OFH        STRIP HIGH NIBBLE
ASCO      CALL      ASC1       CONVERSION ROUTINE
          STAX      D          PUT CHAR IN BUFF.
          INX       D
          RET
*
*         THE FOLLOWING ROUTINE CONVERTS A
*         4 BIT VALUE TO AN ASCII CHARACTER
*         (HEX NOTATION) IN ONLY 6 BYTES
*         (WHO SAYS THE 8080 AIN'T A GOOD
*         PROCESSOR?)
*
ASC1      ADI       90H        A TO F WILL CAUSE
          DAA                  CARRY TO BE SET
          ACI       40H        ADD CARRY AND
          DAA                  ADJUST
          RET
--
```

code in the table of mnemonics and inserts the appropriate mnemonic into the buffer. The data that follows two-byte instructions is then placed into its position in the mnemonic.

In the case of three-byte instructions, operation is similar, except the two bytes of data are reversed when placed in the mnemonic column so that the high byte will come first (the 8080 stores two-byte data low-high).

As an added touch, the letter Q is printed after single-byte data in the mnemonic column to designate octal; the letter A after two-byte data designates split octal. Comments are then printed as described earlier.

Operation is as foolproof as possible. After printing its title, DIS-8 asks for a starting address, which must be entered using split octal notation. The input routine will not accept any digit that is not an octal number, nor will it accept a byte greater than 377Q. An illegal entry will cause a question mark to be printed, and the input register will be cleared, requiring the user to start over.

Each byte entered must consist of three digits, but for an address less than 1000A, only one byte is needed. A space is printed after each byte entered, and if more than two are entered, the last two are used as the starting address. The entry is terminated and disassembly starts when a carriage return is entered.

DIS-8 disassembles for ten lines and waits for a response. If an escape is entered, the user is asked for a new starting address. This makes it possible to follow jumps and calls in the program being disassembled. If any other key is pressed, disassembly continues at the current address.

Modifications

Even though the H8 and its software use an octal number system, I originally wrote DIS-8 with hexadecimal input and output, as shown in the sample run in Listing 3. If you prefer the hex version, the routines labeled BIN, ADDR, PADDR and ASC must be replaced with those shown in Listing 4.

In addition, the following changes must be made:
BUFF + 9 to BUFF + 6
BUFF + 50 to BUFF + 40
BUFF + 25 to BUFF + 20
MVI A,2AH to MVI A,3BH
MVI A,51H and MVI A,41H to MVI A,48H
MVI D,54 to MVI D,44
DS 55 to DS 45.

The hex version's input routine will only accept valid hex numbers, and no leading zeros are required for addresses less than 1000H.

With patience and a little luck, you can decipher a machine-code program using DIS-8. I now have Control BASIC operating on a homemade ROM board in my H8. ■

The complete assembly source for DIS-8 (both versions) is available on cassette from the author for $5, in Heath's TED-8 format.

Listing 1. DIS-8 disassembler for Heath H8.

```
                          *      DIS-8  (OCTAL VERSION)
                          *      AN 8080 DISASSEMBLER
                          *      FOR HEATH H8 COMPUTERS
                          *
                          *      WRITTEN BY PATRICK SWAYNE  11-11-78
                          *      REVISED  3-27-79
                          *      OCTAL VERSION WRITTEN 5-3-79
                          *
040.100                          ORG       040100A
                          *
                          *      MINI CONSOLE DRIVER
                          *      FOR HEATH H8 COMPUTERS
                          *
040.100  006 116                 MVI       B,116Q    FOR 1 STOP BIT
040.102  303 107 040             JMP       UART
```

```
040.105  006 316              MVI    B,316Q    FOR 2 STOP BITS
040.107  076 201      UART    MVI    A,201Q    SET UP 8251
040.111  323 373              OUT    373Q
040.113  076 100              MVI    A,100Q
040.115  323 373              OUT    373Q
040.117  177                  MOV    A,A       DELAY TACTIC
040.120  170                  MOV    A,B       SET STOP BITS
040.121  323 373              OUT    373Q
040.123  076 025              MVI    A,025Q    ENABLE TRANSMIT
040.125  323 373              OUT    373Q      AND RECIEVE
040.127  303 162 040          JMP    DIS
040.132  333 373      IN      IN     373Q      READY TO
040.134  346 002              ANI    2         RECIEVE CHAR?
040.136  312 132 040          JZ     IN        NO, TRY AGAIN
040.141  333 372              IN     372Q      YES, GET IT
040.143  346 177              ANI    177Q
040.145  311                  RET
040.146  365          OUT     PUSH   PSW       SAVE ACC
040.147  333 373      OUT1    IN     373Q      READY TO
040.151  346 001              ANI    1         TRANSMIT CHAR?
040.153  312 147 040          JZ     OUT1      NO, TRY AGAIN
040.156  361                  POP    PSW       YES, RESTORE ACC
040.157  323 372              OUT    372Q      AND TRANSMIT
040.161  311                  RET
                      *
                      *       MAIN PROGRAM
                      *
040.162  315 275 041  DIS     CALL   PLF
040.165  315 210 041          CALL   CLRBF
040.170  041 074 042          LXI    H,TITLE   GET TITLE
040.173  315 061 042          CALL   PSTRG     PRINT IT
040.176  315 275 041          CALL   PLF
040.201  315 275 041  DIS0    CALL   PLF
040.204  041 115 042          LXI    H,QADDR   ASK START ADDR
040.207  315 061 042          CALL   PSTRG
040.212  315 320 041          CALL   ADDR      GET IT
040.215  315 275 041          CALL   PLF
040.220  016 012      DIS1    MVI    C,10      10 LINE PAGE
040.222  315 275 041          CALL   PLF
040.225  315 007 042  DIS2    CALL   PADDR     PRINT ADDRESS
040.230  106                  MOV    B,M       GET INSTRUCTION
040.231  345                  PUSH   H         SAVE ADDRESS
                      *
                      *       PROCESS 1, 2, AND 3 BYTE INSTRUCTIONS
                      *
040.232  041 001 052          LXI    H,TABL1   IS IT 2 BYTE
040.235  176          DIS3    MOV    A,M       INSTRUCTION?
040.236  376 377              CPI    0FFH
040.240  312 253 040          JZ     DIS4      NO, CHECK IF
040.243  270                  CMP    B         3 BYTES
040.244  312 323 040          JZ     DIS6      2 BYTE INST.
040.247  043                  INX    H
040.250  303 235 040          JMP    DIS3
040.253  043          DIS4    INX    H         IS IT 3 BYTE
040.254  176                  MOV    A,M       INSTRUCTION?
040.255  376 377              CPI    0FFH
040.257  312 271 040          JZ     DIS5      NO, 1 BYTE
040.262  270                  CMP    B
040.263  312 374 040          JZ     DIS7      YES, PRINT REST
040.266  303 253 040          JMP    DIS4      OF INSTRUCTION
                      *
                      *       SINGLE BYTE INSTRUCTIONS
                      *
040.271  341          DIS5    POP    H         RESTORE ADDR
040.272  345                  PUSH   H         SAVE AGAIN
040.273  021 173 042          LXI    D,BUFF+9  POINT TO BUFFER
040.276  176                  MOV    A,M       GET BYTE
040.277  315 023 042          CALL   ASC       PUT IN BUFFER
040.302  315 114 041          CALL   MNEM      PUT IN MNEMONIC
040.305  021 244 042          LXI    D,BUFF+50 PUT * IN BUFF
040.310  076 052              MVI    A,2AH
040.312  022                  STAX   D
040.313  023                  INX    D
040.314  341                  POP    H         RESTORE ADDR
040.315  315 243 041          CALL   COM2      PRINT COMMENT
040.320  303 056 041          JMP    PAGE      CK END OF PAGE
                      *
                      *       2 BYTE INSTRUCTIONS
                      *
040.323  341          DIS6    POP    H         RESTORE ADDR
040.324  345                  PUSH   H         SAVE AGAIN
040.325  021 173 042          LXI    D,BUFF+9  POINT TO BUFFER
040.330  176                  MOV    A,M       GET BYTE
040.331  315 023 042          CALL   ASC       PUT IT IN BUFF
040.334  023                  INX    D         SPACE
040.335  043                  INX    H         NEXT BYTE
040.336  176                  MOV    A,M
040.337  315 023 042          CALL   ASC       PUT IT IN BUFF
040.342  315 114 041          CALL   MNEM      PRINT MNEMONIC
040.345  341                  POP    H
040.346  043                  INX    H         PRINT NEXT BYTE
040.347  176                  MOV    A,M       OF 2 BYTE INST.
040.350  315 023 042          CALL   ASC
040.353  076 121              MVI    A,51H     PRINT Q
040.355  022                  STAX   D
040.356  021 244 042          LXI    D,BUFF+50
040.361  076 052              MVI    A,2AH     PRINT *
```

```
040.363  022                     STAX    D
040.364  023                     INX     D
040.365  053                     DCX     H          BACK UP
040.366  315 235 041             CALL    COM1       PRINT COMMENT
040.371  303 056 041             JMP     PAGE       CK END OF PAGE
                         *
                         *       3 BYTE INSTRUCTIONS
                         *
040.374  341             DIS7    POP     H          RESTORE ADDR
040.375  345                     PUSH    H          SAVE AGAIN
040.376  021 173 042             LXI     D,BUFF+9   POINT TO BUFF
041.001  176                     MOV     A,M        GET BYTE
041.002  315 023 042             CALL    ASC        PUT IT IN BUFF
041.005  023                     INX     D          SPACE
041.006  043                     INX     H          NEXT BYTE
041.007  176                     MOV     A,M
041.010  315 023 042             CALL    ASC        PUT IT IN BUFF
041.013  023                     INX     D
041.014  043                     INX     H
041.015  176                     MOV     A,M
041.016  315 023 042             CALL    ASC        3RD BYTE IN BUFF
041.021  315 114 041             CALL    MNEM       PRINT MNEMONIC
041.024  341                     POP     H
041.025  043                     INX     H
041.026  043                     INX     H          PRINT NEXT
041.027  176                     MOV     A,M        2 BYTES OF
041.030  315 023 042             CALL    ASC        3 BYTE INST.
041.033  053                     DCX     H
041.034  176                     MOV     A,M
041.035  315 023 042             CALL    ASC        PRINT NXT BYTE
041.040  076 101                 MVI     A,41H      PRINT LETTER A
041.042  022                     STAX    D
041.043  021 244 042             LXI     D,BUFF+50
041.046  076 052                 MVI     A,2AH      PRINT *
041.050  022                     STAX    D
041.051  023                     INX     D
041.052  053                     DCX     H
041.053  315 227 041             CALL    COM        PRINT COMMENTS
                         *
                         *       END OF PAGE CHECK
                         *
041.056  345             PAGE    PUSH    H
041.057  041 162 042             LXI     H,BUFF     PRINT CONTENTS
041.062  315 061 042             CALL    PSTRG      OF BUFFER
041.065  315 210 041             CALL    CLRBF
041.070  341                     POP     H
041.071  315 275 041             CALL    PLF
041.074  043                     INX     H          GET NEXT INST.
041.075  015                     DCR     C          END OF PAGE?
041.076  302 225 040             JNZ     DIS2       NO, CONTINUE
041.101  315 132 040             CALL    IN         YES, WANT TO
041.104  376 033                 CPI     1BH        CONTINUE?
041.106  312 201 040             JZ      DIS0       NO, BEGIN PGM
041.111  303 220 040             JMP     DIS1       YES, START PAGE
                         *
                         *       PRINT MNEMONICS
                         *
041.114  041 251 042     MNEM    LXI     H,TABLE    LOOK UP OPCODE
041.117  176             MNEMO   MOV     A,M        IN TABLE
041.120  270                     CMP     B          IS THIS IT?
041.121  312 137 041             JZ      MNEM2      YES, PUT IN BUFF
041.124  043             MNEM1   INX     H          NO, GET NEXT
041.125  176                     MOV     A,M        OPCODE
041.126  376 004                 CPI     4
041.130  302 124 041             JNZ     MNEM1
041.133  043                     INX     H
041.134  303 117 041             JMP     MNEMO      TRY AGAIN
041.137  043             MNEM2   INX     H          SKIP OVER OPCODE
041.140  021 213 042             LXI     D,BUFF+25  POINT TO BUFFER
041.143  176                     MOV     A,M        GET FIRST CHAR.
041.144  022             MNEM3   STAX    D          PUT IT IN BUFF
041.145  043                     INX     H          GET NEXT CHAR
041.146  023                     INX     D
041.147  176                     MOV     A,M
041.150  376 010                 CPI     8          CHECK FOR SPACES
041.152  314 175 041             CZ      SPACE
041.155  376 007                 CPI     7
041.157  314 175 041             CZ      SPACE
041.162  376 006                 CPI     6
041.164  314 175 041             CZ      SPACE
041.167  376 004                 CPI     4          END OF MNEM.?
041.171  302 144 041             JNZ     MNEM3      NO, CONTINUE
041.174  311                     RET                YES, RETURN
                         *
                         *       PRINT SPACES
                         *
041.175  305             SPACE   PUSH    B
041.176  117                     MOV     C,A        NO OF SPACES
041.177  023             SPACE1  INX     D          "PRINT" A SPACE
041.200  015                     DCR     C          END OF SPACES?
041.201  302 177 041             JNZ     SPACE1     NO, CONTINUE
041.204  301                     POP     B          YES,
041.205  043                     INX     H          GET NEXT CHAR
041.206  176                     MOV     A,M
041.207  311                     RET                RETURN
                         *
                         *       CLEAR BUFFER
```

```
                          *
041.210   026 066         CLRBF    MVI     D,54        SIZE OF BUFFER - 1
041.212   041 162 042              LXI     H,BUFF
041.215   066 040         CLRB1    MVI     M,20H       PUT SPACE IN BUFFER
041.217   043                      INX     H
041.220   025                      DCR     D
041.221   302 215 041              JNZ     CLRB1
041.224   066 004                  MVI     M,4
041.226   311                      RET
                          *
                          *        PRINT ASCII COMMENTS
                          *
041.227   176             COM      MOV     A,M         GET 1ST BYTE
041.230   315 244 041              CALL    COMA        PRINT IT
041.233   043                      INX     H
041.234   023                      INX     D
041.235   176             COM1     MOV     A,M         GET BYTE
041.236   315 244 041              CALL    COMA        PRINT IT
041.241   043                      INX     H
041.242   023                      INX     D
041.243   176             COM2     MOV     A,M         GET BYTE
041.244   346 177         COMA     ANI     7FH         NO PARITY
041.246   365                      PUSH    PSW         SAVE CHAR
041.247   326 137                  SUI     5FH         SMALL LETTER?
041.251   372 261 041              JM      COMA1       NO
041.254   361                      POP     PSW         YES, CAPITALIZE
041.255   346 137                  ANI     5FH
041.257   022                      STAX    D           PUT IT IN BUFF
041.260   311                      RET
041.261   361             COMA1    POP     PSW
041.262   365                      PUSH    PSW
041.263   326 041                  SUI     21H         CONTROL CHARS?
041.265   362 272 041              JP      COMA2
041.270   361                      POP     PSW         IF SO, DON'T
041.271   311                      RET                 PRINT THEM
041.272   361             COMA2    POP     PSW
041.273   022                      STAX    D           PUT IT IN BUFF
041.274   311                      RET
                          *
                          *        PRINT CR AND LF
                          *
041.275   345             PLF      PUSH    H           SAVE H,L
041.276   041 153 042              LXI     H,CRLF      GET CR, LF
041.301   315 061 042              CALL    PSTRG       PRINT THEM
041.304   341                      POP     H           RESTORE H,L
041.305   311                      RET
                          *
                          *        ASCII TO BINARY CONVERSION ROUTINE
                          *
041.306   326 060         BIN      SUI     30H         NO ENTRY < 0
041.310   330                      RC
041.311   306 370                  ADI     30H-38H     NO ENTRY > 7
041.313   330                      RC
041.314   306 010                  ADI     8           ADJUST
041.316   267                      ORA     A           CLEAR ERROR FLAG
041.317   311                      RET
                          *
                          *        GET STARTING ADDRESS
                          *
041.320   041 000 000     ADDR     LXI     H,0         CLEAR H,L
041.323   257             ADDR0    XRA     A           CLEAR A
041.324   006 003                  MVI     B,3         SET DIGIT COUNTER
041.326   117             ADDR1    MOV     C,A         SAVE A
041.327   315 132 040              CALL    IN          GET CHARACTER
041.332   315 146 040              CALL    OUT         ECHO
041.335   376 015                  CPI     0DH         DONE?
041.337   310                      RZ                  IF SO, RETURN
041.340   315 306 041              CALL    BIN         CONVERT TO BINARY
041.343   332 376 041              JC      ERR         BAD ENTRY
041.346   201                      ADD     C           ADD PREVIOUS ENTRIES
041.347   005                      DCR     B           3 DIGITS ENTERED?
041.350   312 364 041              JZ      ADDR2       YES, PROCESS BYTE
041.353   027                      RAL                 NO, MOVE ENTRY
041.354   027                      RAL                 UP THREE PLACES
041.355   027                      RAL
041.356   332 376 041              JC      ERR         NUMBER TOO LARGE
041.361   303 326 041              JMP     ADDR1       GET ANOTHER ENTRY
041.364   145             ADDR2    MOV     H,L         MOVE LAST BYTE UP
041.365   157                      MOV     L,A         NEW BYTE IN L
041.366   076 040                  MVI     A,20H       SPACE BETWEEN BYTES
041.370   315 146 040              CALL    OUT
041.373   303 323 041              JMP     ADDR0       GET ANOTHER BYTE
                          *
                          *        ERROR ROUTINE
                          *
041.376   041 147 042     ERR      LXI     H,QUES      GET QUESTION MARK
042.001   315 061 042              CALL    PSTRG       PRINT IT
042.004   303 320 041              JMP     ADDR        TRY AGAIN
                          *
                          *        PRINT ADDRESS
                          *
042.007   174             PADDR    MOV     A,H         GET ADDR HI BYTE
042.010   021 162 042              LXI     D,BUFF
042.013   315 023 042              CALL    ASC         PRINT IT
042.016   076 056                  MVI     A,'.'       PRINT A PERIOD
042.020   022                      STAX    D
042.021   023                      INX     D
```

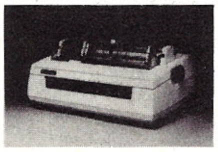
```
042.022  175                        MOV    A,L        GET LOW BYTE
                          *         JMP    ASC        PRINT IT
                          *
                          *         BINARY TO ASCII CONVERSION ROUTINE
                          *
042.023  365              ASC       PUSH   PSW        SAVE BYTE
042.024  017                        RRC               MOVE MOST
042.025  017                        RRC               SIGNIFICANT BITS
042.026  017                        RRC               DOWN
042.027  017                        RRC
042.030  017                        RRC
042.031  017                        RRC
042.032  346 003                    ANI    3          STRIP OTHER BITS
042.034  315 054 042                CALL   ASCO       CONVERT TO ASCII
042.037  361                        POP    PSW        RESTORE BYTE
042.040  365                        PUSH   PSW        SAVE IT
042.041  017                        RRC               MOVE 3 MIDDLE BITS
042.042  017                        RRC               DOWN
042.043  017                        RRC
042.044  346 007                    ANI    7          STRIP OTHER BITS
042.046  315 054 042                CALL   ASCO       CONVERT TO ASCII
042.051  361                        POP    PSW        RESTORE BYTE
042.052  346 007                    ANI    7          STRIP HIGH BITS
042.054  306 060          ASCO      ADI    30H        ADD ASCII BIAS
042.056  022                        STAX   D          PUT CHAR IN BUFF
042.057  023                        INX    D
042.060  311                        RET
                          *
                          *         STRING PRINTING ROUTINE
                          *
042.061  176              PSTRG     MOV    A,M        POINT TO STRING
042.062  376 004                    CPI    4          END OF STRING?
042.064  310                        RZ                YES, RETURN
042.065  315 146 040                CALL   OUT        NO, PRINT CHAR
042.070  043                        INX    H          MOVE TO NEXT CHAR
042.071  303 061 042                JMP    PSTRG
                          *
                          *         STRINGS
                          *
042.074  104 111 123      TITLE     DB     'DIS-8, VERS. 3.0'
042.114  004                        DB     4
042.115  105 116 124      QADDR     DB     'ENTER STARTING ADDRESS:  '
042.146  004                        DB     4
042.147  040 077 040      QUES      DB     ' ? ',4
042.153  015 012 000      CRLF      DB     0DH,0AH,0,0,0,0,4
042.162                   BUFF      DS     55
                          *
                          *         TABLE OF MNEMONICS
                          *
042.251  000              TABLE     DB     00
042.252  116 117 120                DB     'NOP'
042.255  004                        DB     4
042.256  001                        DB     01
042.257  114 130 111                DB     'LXI',7,'B,'
042.265  004                        DB     4
042.266  002                        DB     02
042.267  123 124 101                DB     'STAX',6,'B'
042.275  004                        DB     4
042.276  003                        DB     03
042.277  111 116 130                DB     'INX',7,'B'
042.304  004                        DB     4
042.305  004                        DB     04
042.306  111 116 122                DB     'INR',7,'B'
042.313  004                        DB     4
                          *
                          *         THE TABLE CONTINUES FOR THE REST
                          *         OF THE OP CODES, BUT WILL NOT BE
                          *         LISTED HERE TO SAVE SPACE.
                          *
                          *         TABLE OF 2 AND 3 BYTE OP CODES.
                          *
                                    LON    L          RESUME LISTING
052.001  006              TABL1     DB     06H
052.002  016                        DB     0EH
052.003  026                        DB     16H
052.004  036                        DB     1EH
                          *
                          *         AND SO ON FOR THE REST OF THE
                          *         2 BYTE INSTRUCTIONS, UNTIL...
                          *
                                    LON    L          RESUME LISTING
052.023  377                        DB     0FFH
                          *
                          *         OFF SIGNIFIES END OF 2 BYTE CODES.
                          *         THEN 3 BYTE CODES ARE TABLED.
                          *
052.024  001                        DB     01H
052.025  021                        DB     11H
052.026  041                        DB     21H
052.027  042                        DB     22H
                          *
                          *         AND SO ON, AGAIN ENDING IN OFF.
                          *
                                    LON    L          RESUME LISTING
052.057                             END    040100A
```

WE WILL NOT BE UNDERSOLD

16K MEMORY UPGRADE KITS $54

for TRS-80*, Apple II, Sorcerer (specify)

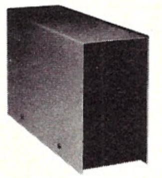
PRINTERS



NEC Spinwriter

Letter Quality High Speed Printer
Includes TRS-80* interface software, quick change print fonts, 55 cps, bidirectional, high resolution plotting, graphing, proportional spacing **$2689**
With Tractor Feed **$2889**

DIABLO 1650 R.O. $2890	KSR $3285
779 CENTRONICS TRACTOR FEED PRINTER Same as Radio Shack line printer I	$969
737 CENTRONICS FRICTION & PIN FEED PRINTER 9 × 7 matrix	$839
730 CENTRONICS FRICTION & PIN FEED PRINTER 7 × 7 matrix Same as Radio Shack line printer II	$639
P1 CENTRONICS PRINTER Same as Radio Shack quick printer	$269
PAPER TIGER (IP440) Includes 2K buffer and graphics option	$939
TI-810 Faster than Radio Shack line printer III — Parallel and serial w/TRS-80* interface software	$1575
with upper and lower case and paper tray	$1665
OKIDATA Microline 80 Friction and pin feed	$559
Tractor Feed, friction, and pin feed	$679
EATON LRC 7000 + 64 columns, plain paper	$349
ANADEX DP-9500	$1389
CAT MODEM Originate and answer same as Radio Shack Telephone Interface II	$148
LEEDEX MONITOR Video 100	$129



DISK DRIVES $314

More capacity than Radio Shack 35 Track (80 K Bytes) drives. Fully assembled and tested. Ready to plug-in and run the moment you receive it. Can be intermixed with each other and Radio Shack drive on same cable. TRS-80* compatible silver enclosure.

90 DAY WARRANTY. ONE YEAR ON POWER SUPPLY.

FOR TRS-80*		
CCI-100	5¼", 40 Track (102K Bytes) for Model I	**$314**
CCI-200	5¼", 77 Track (197K Bytes) for Model I	**$549**
CCI-800	8" Drive for Model II (½ Meg Bytes)	**$795**
For Zenith Z89		
CCI-189	5¼", 40 Track (102K Bytes) add-on drive	**$394**
Z-87	Dual 5¼" add-on drive system	**$995**
DISKETTES —	Box of 10 (5¼") - with plastic library case	**$24.95**
	8" double density for Model II (box of 10)	**$36.49**

COMPLETE SYSTEMS

TRS-80* LEVEL II-16K with keypad	$719
TRS-80* Expansion Interface	$269
ZENITH Z89, 48K all-in-one computer	$2595
ZENITH Z19	$740
TELEVIDEO 912B $745	920B $769
ATARI 400 $489 **ATARI 800 $799**	**TI 99/4 $894**
MATTEL INTELLIVISION	$249
NORTH STAR Horizon 1 32K, Double Density	$2129

DISK OPERATING SYSTEMS

PATCHPAK #4 by Percom Data	$ 8.95
CP/M for Model I, Zenith **$145** • for Model II, Altos	$169.00
NEWDOS Plus — with over 200 modifications 35track	$ 89.00
and corrections to TRS-DOS 40 or 70 Track	$ 99.00

SOFTWARE FOR THE TRS-80* Software w/Manual / Manual Alone

CCI-INVESTMENT PORTFOLIO MANAGER: This is what investors have been waiting for! This powerful program was developed by security analysts working with software designers. It comes on one cassette — 16K LEVEL II BASIC on one side, 32K DISK BASIC on the other. Store and report data; Review your portfolio; Produce detailed status, value, gain, and security analysis; Compare alternatives **$49.95/$10**

INTELLIGENT TERMINAL SYSTEM ST-80-III BY LANCE MIKLUS: Enables a TRS-80* to act as a dial-up terminal on any standard time sharing network. Provides a TRS-80* with control key, ESC Key, Repeat Key, Rub Out Key, Break Key, full upper and lower case support, selectable printer output and program selectable transmission rates **$139/$10**

CCA-DATA MANAGEMENT SYSTEM: Automate your information processing tasks. You can create a file of customer information; quickly and easily add, delete or update records; search a file; keep a file in order of the value in any field; and print records and labels in any desired sequence or from just a part of a file. Requires 32K TRS-80 and one drive. **$74.95/$10**

CSA-MAILIST SYSTEM: Creates, maintains and efficiently utilizes a name, address and telephone number file. 400 individual name/address entries can be maintained on a single density mini-floppy, and are manipulated directly by record number (direct access file method). Sorts can be performed, name + address combinations can be coded. Listing-directories and labels can be printed. A conversion facility is provided to convert most sequential name, address file formats into direct. Requires 32K TRS-80 and one drive. **$49.95/$10**

S & M SYSTEMS
INSEQ-80™: Indexed Sequential Access Method (ISAM) for the TRS-80 Model I. A must for anyone writing business programs. Eliminate wasted disk space from direct record processing. Split second access to any record. Access data records instantly via alpha/numeric "key" eg. Part NR, zip code or sequentially in ascending key sequence. Add/modify records in any order. Access up to three files per program — Files may be spread over multiple disks. Machine language processing from your basic program. Utility program to convert direct files to INSEQ-80 format. **$49.95/$10**

FULLY INTERACTIVE ACCOUNTING PACKAGE: ISAM (INSEQ-80) based. Includes General Ledger, Accounts Payable, Accounts Receivable and Payroll. System runs "stand alone" or "co-ordinated G/L" at users option. Based on Osborne accounting method. Requires 32K, TRS-80, 2 or 3 drives. N/A CA.

General Ledger	**$99/$10**
Accounts Receivable	**$99/$10**
Accounts Payable	**$99/$10**
Payroll	**$99/$10**

Osborne books: Req'd as additional documentation $20 ea

INVENTORY Requires 32K, TRS-80, 1 drive **$125/$10**
INSORT-80: Callable form BASIC via USR. Sorts "Random" Disk Files. "Disk" to "Disk" sort times — 350 records in 35 secs, 1000 records in 6 minutes, 3500 records in 12 minutes. Machine language processing. Up to 35 sort keys ascending/descending. Utility to build BASIC program. Runs under NEWDOS. **$49.95/$10**

ACCESSORIES
HEAD CLEANING DISKETTE: Cleans drive Read/Write head in 30 seconds. Diskette absorbs loose oxide particles, fingerprints, and other foreign particles that might hinder the performance of the drive head. Lasts at least 3 months with daily use. Specify 5¼" or 8". **$20 ea/$45 for 3**

FLOPPY SAVER: Protection for center holes of 5¼" floppy disks. Only 1 needed per diskette. Kit contains centering post, pressure tool, tough 7-mil mylar reinforcing rings. Installation tools and rings for 25 diskettes. **$11.95**
Re-orders of rings only: **$ 7.95**

CP/M BASED SOFTWARE for Zenith, Altos, Radio Shack, Apple Software w/Manual / Manual Alone

MICROSOFT
BASIC-80: Disk Extended BASIC ANSI compatible with long variable names, WHILE/WEND, chaining, variable length file records. **$325/$25**
BASIC COMPILER: Language compatible with BASIC-80 and 3-10 times faster execution. Produces standard Microsoft relocatable binary output. Includes MACRO-80 Also linkable to FORTRAN-80 or COBOL-80 code modules. **$350/$25**
FORTRAN-80: ANSI 66 (except for COMPLEX) plus many extensions. Includes relocatable object compiler, linking loader, library with manager. Also includes MACRO-80 (see below) **$425/$25**
COBAL-80: Level 1 ANSI '74 standard COBAL plus most of Level 2. Full sequential, relative and indexed file support with variable file names. STRING, UNSTRING, COMPUTE, VARYING/UNTIL, EXTEND. CALL, COPY, SEARCH, 3-dimensional arrays, compound and abbreviated conditions, nested IF. Powerful interactive screen-handling extensions. Includes compatible assembler, linking loader, and relocatable library manager as described under MACRO-80. **$700/$25**

Z-80 SOFTCARD FOR APPLE: Your key to future software expansion. Get the best of both worlds, Apple's 6502 and CP/M Z-80. Plug in the card and get a Z80. Supports Apple language card and all Apple peripherals. Comes with set of three manuals. **$339/$75**

CCI-TELNET VERSION 5: A communication Package which enables microcomputer users to communicate both with Large Mainframes and other microcomputers. Extensive commands make it useful in many applications where communication between computers is necessary. Powerful terminal mode enabling user to save all data from a session on disk. Completely CP/M compatible. Multiple communication protocols supported. Able to transfer files in both directions without protocol where the other machine does not support any protocol. Extensive ON-SCREEN help. Source code provided. **$149/$15**

MICROPRO-WORD-STAR: Menu driven visual word processing system for use with standard terminals. Text formatting performed on screen. Facilities for text paginate, page number, justify, center and underscore. User can print one document while simultaneously editing a second. Edit facilities include global search and replace. Read/Write to other text files, block move, etc. Requires CRT terminal with addressable cursor positioning. **$399/$40**

BDS 'C' COMPILER: Supports most features of language, including structures, arrays, pointers, recursive function evaluation, and overlays. Package contains: compiler, linker, library manager; sample source files include games, a terminal emulator with disk I/O plus the source for many standard library functions; BDS C User's Guide; Book - The C Programming Language by Dennis Ritchie and Brian Kernigham. Requires at least 24K of RAM **$125/$20**

CONFIGURABLE BUSINESS SYSTEM BY DMA: CBS is a data management system that allows true transaction processing. Custom accounting systems for payables, receivables, inventory control, order entry, and general ledger can be set up without using any programming languages. CBS can be used to define an application such as an inventory control system by specifying master files to describe the inventory, customer and vendor files. Transaction files can then be used to describe activities such as purchases and sales. An extremely easy-to-use data entry program is used to enter information about customers, vendors, inventory, sales and purchases. After data entry is complete, an update program can process the transactions against the various master files, updating account balances and inventory quantities. The system features a screen menu generator and a comprehensive report generator which can be used to produce invoices, purchase orders, re-order reports, mailing labels or other special reports specific to the application. Good documentation and a demonstration inventory system supplied. Requires at least 48K memory. Does not require any support language. **$295**

DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED — **Send for FREE Catalogue**

The CPU SHOP

✓256

TO ORDER CALL TOLL FREE 1-800-343-6522
Massachusetts residents call **(617) 242-3361**
For detailed technical information, call 617/242-3361
Hours: 10AM-6PM (EST) M-F (Sat. till 5)
*TRS-80 is a Tandy Corporation Trademark

5 Dexter Row, Dept. K8M
Charlestown, Massachusetts 02129

Massachusetts residents add 5% sales tax

Quantities on some items are limited

VISA® · master charge

PET I/O Port Expander

Series on enhancing the PET's capabilities continues with a description of joystick interfacing.

William F. Pytlik
9012 Maritime Court
Springfield, VA 22153

Many games and interactive computer programs can become more pleasant and efficient to use with the addition of joysticks. For complex, high-performance computers, a full range of freedom joysticks is appropriate; but for home computers, joysticks have four switches that move the cursor in four directions. These joysticks often do not provide a method to directly control the cursor in the diagonal direction.

Hardware

You can readily implement professional joysticks that control the cursor in a diagonal direction with the addition of four 100k Ohm potentiometers (Photo 1). These are readily available through many of the electronic magazine advertisers at approximately $5 each.

You need a simple 3 × 4 inch printed circuit board to interface the joysticks via the "Expander" (June 1980, p. 58) to the computer I/O port. Fig. 1 presents the full-size pattern of the printed circuit board. If the joysticks are wired as indicated in Fig. 2, the resistances in Table 1 will be observed at the four potentiometers of the joysticks.

The PET I/O port, when programmed for input, will essentially ignore a 100k Ohm input and remain high (1). If the input is grounded, then the I/O port will recognize this as a 0. Consequently, peeking the input port will result in the decimal values shown in Table 2.

The joysticks can be housed in any case of appropriate size. I chose to use a 5 × 3 × 2 inch aluminum box. Photo 2 shows the completed joysticks. The switch on the box is used as a "FIRE" button, useful for many games. Fig. 2 illustrates how these switches are tied into the joystick system. Essentially, when depressed, the switch brings all four inputs to the computer to ground (binary 0000 left, 0000 right or decimal 0 left, 0 right). Any program using joysticks as inputs and requiring a "FIRE" in-

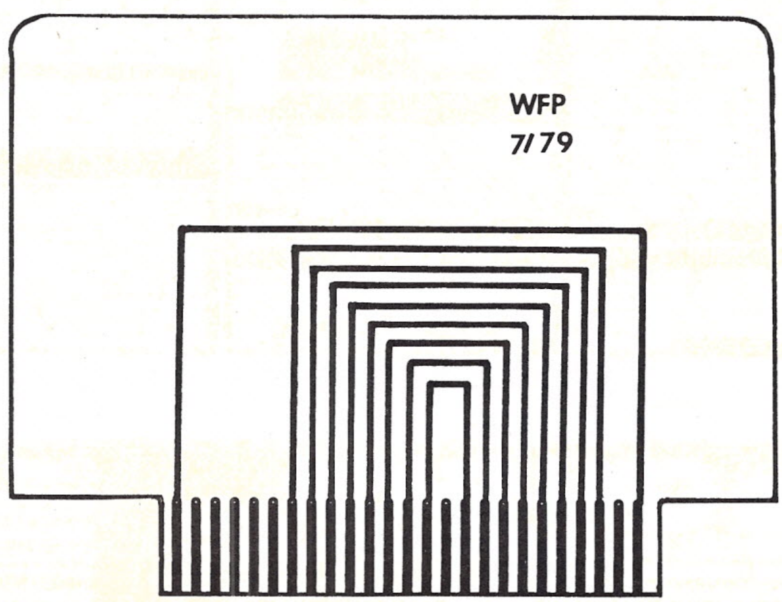


Fig. 1. Printed circuit board (full size).

Position (In degrees)	Resistance Potentiometer 1	2	3	4
0	100k	100k	100k	0
45	0	100k	100k	0
90	0	100k	100k	100k
135	0	0	100k	100k
180	100k	0	100k	100k
225	100k	0	0	100k
270	100k	100k	0	100k
315	100k	100k	0	0

Table 1.

Position (degrees)	Left Joystick	Right Joystick
0	224	14
45	96	6
90	112	7
135	48	3
180	176	11
225	144	9
270	208	13
315	192	12

Table 2.

2	—Joysticks w/four potentiometers
1	—6 foot DIN cable. Cut in half.
D1-D4	—1N914
S1,S2	—N.O. MOM contact switch
Misc.	—Hardware, Aluminum Box, Wire

Table 3. Parts list.

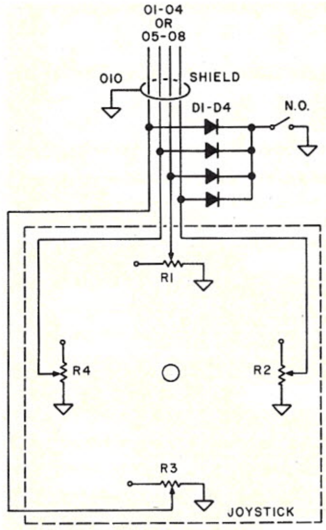


Fig. 2. Wiring the joystick.

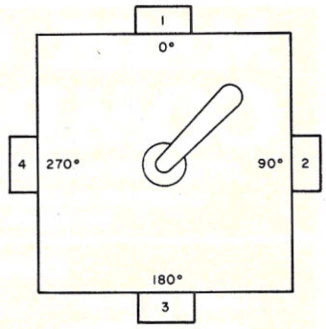


Fig. 3.

dication would simply test for these values.

Software

Having completed the joysticks, you are now ready to enter Listing 1, which contains a simple BASIC program to maneuver two cursors across the screen and a children's game, called "YOU'RE IT." Although the program listing is self-documenting, you should note that to eliminate wrap-around, change lines 3000-3050 and 4000-4050 to the lines in Listing 2.

The cursors defined in line 1030 look the same. If different-looking cursors are required, different values must be poked to screen memory; a solid white cursor is screen value 160. To speed up cursor movement, the programs must be written in machine language. Additionally, the "FIRE" switch in this case could be used to change the speed of cursor movement.

Conclusion

A joystick is invaluable for serious interactive computer work. The joystick controller in this article is more intricate than the video game joysticks, but less complex than those used on expensive computers. They are versatile and low in cost. The simple BASIC program presented in this article illustrates the simplest method to control the cursors via joysticks; the game, although very basic, provides hours of enjoyment for children.■

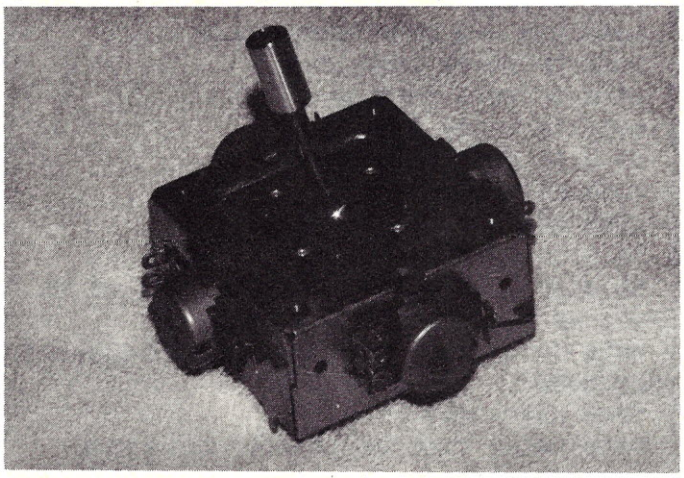


Photo 1. Potentiometers used in the PET joystick.

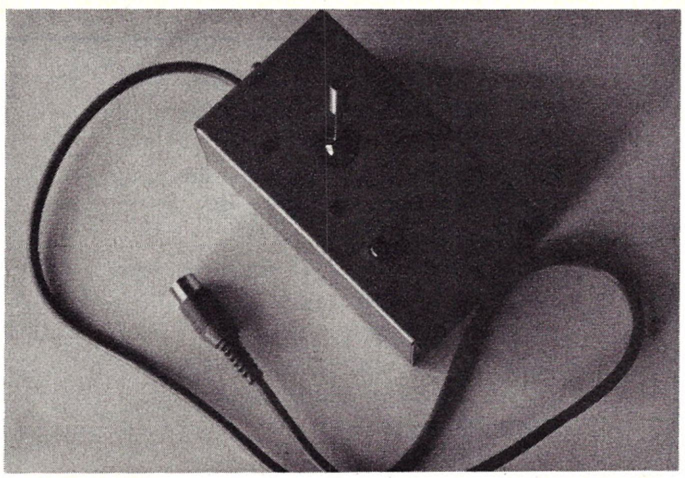


Photo 2. Completed joystick.

Listing 1. Program for the joystick.

```
5 REM *** JOYSTICK ROUTINE ***
6 PRINT"♥"
10 REM **OPEN USER PORT FOR READ**
20 POKE 59459,0
30 REM **INITIALIZE CURSORS**
40 XL=0:YL=0:XR=39:YR=24
50 GOSUB 1000
60 REM **LOOK AT LEFT JOYSTICK**
70 A=PEEK(59471) AND 240
80 REM **DETERMINE POSITION**
90 GOSUB 2000
100 XL=XL+X:YL=YL+Y
110 REM ** WRAP AROUND**
120 GOSUB 3000
130 REM **LOOK AT RIGHT JOYSTICK**
140 A=PEEK(59471) AND 15
150 REM **DETERMINE POSITION**
160 GOSUB 2000
170 XR=XR+X:YR=YR+Y
180 REM **WRAP AROUND**
190 GOSUB 4000
200 REM **DISPLAY CURSORS**
205 POKE Z,32:POKE Z1,32
210 GOSUB 1000
220 REM **GAME**
230 GOSUB 5000
240 GOTO 70
250 END
```

```
1000 REM *** PRINT CURSORS **
1010 Z=32768+(YL*40)+XL
1020 Z1=32768+(YR*40)+XR
1030 POKE Z,102:POKE Z1,102
1040 REM **102 IDENTIFIES THE CURSOR**
1050 RETURN
2000 REM *** IDENTIFY POSITION ***
2005 X=0:Y=0
2010 IF A=15 OR A=240 THEN RETURN
2020 IF A=224 OR A=14 THEN Y=Y-1:RETURN
2030 IF A=176 OR A=11 THEN Y=Y+1:RETURN
2040 IF A=112 OR A=7 THEN X=X+1:RETURN
2050 IF A=208 OR A=13 THEN X=X-1:RETURN
2060 IF A=96 OR A=6 THEN Y=Y-1:X=X+1:RE
TURN
2070 IF A=48 OR A=3 THEN Y=Y+1:X=X+1:RE
TURN
2080 IF A=144 OR A=9 THEN Y=Y+1:X=X-1:R
ETURN
2090 IF A=192 OR A=12 THEN Y=Y-1:X=X-1:
RETURN
3000 REM *** WRAP AROUND ***
3010 IF XL>39 THEN XL=0
3020 IF XL<0 THEN XL=39
3030 IF YL>24 THEN YL=0
3040 IF YL<0 THEN YL=24
3050 RETURN
4000 REM *** WRAP AROUND ***
4010 IF XR>39 THEN XR=0
4020 IF XR<0 THEN XR=39
4030 IF YR>24 THEN YR=0
4040 IF YR<0 THEN YR=24
4050 RETURN
5000 REM *** GAME ***
5010 IF Z=Z1 THEN 5050
5020 IF Z=Z1+1 OR Z=Z1-1 OR Z=Z1-41 OR
Z=Z1-40 OR Z=Z1-39 THEN 5050
5030 IF Z=Z1+39 OR Z=Z1+40 OR Z=Z1+41 T
HEN 5050
5040 GOTO 5060
5050 PRINT "YOU'RE IT!!!!":GOTO 250
5060 RETURN
READY.
```

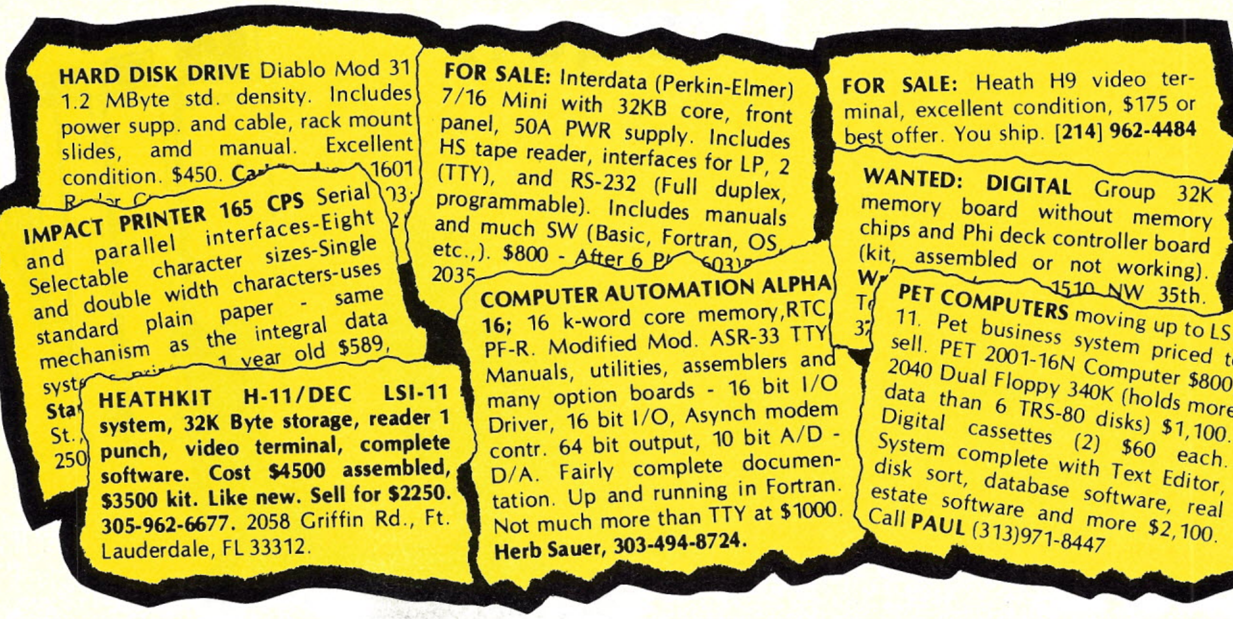
```
3000 REM *** WRAP AROUND ***
3010 IF XL>39 THEN XL=39
3020 IF XL<0 THEN XL=0
3030 IF YL>24 THEN YL=24
3040 IF YL<0 THEN YL=0
3050 RETURN
4000 REM *** WRAP AROUND ***
4010 IF XR>39 THEN XR=39
4020 IF XR<0 THEN XR=0
4030 IF YR>24 THEN YR=24
4040 IF YR<0 THEN YR=0
4050 RETURN
```

Listing 2. Program modification.

Attention

BARGAIN HUNTERS

Receive Hundreds of Classified Ads Like These Every Month



COMPUTER SHOPPER, the new buy, sell, and trade publication, is ready to help you with the latest information on personal, small business and large-system computers, accessories and software.

Each ad-packed issue is full of bargains you are looking for. Included are ads from individuals throughout the United States who are selling their good, pre-owned equipment just so they can trade-up to new equipment coming on the market.

But, COMPUTER SHOPPER'S bargains won't be yours unless you subscribe. This useful, money-saving publication can become your way to communicate with other buyers, sellers, and traders all over the nation.

Whether you are a hobbyist or a part-time user, COMPUTER SHOPPER will put you in touch with the nationwide computer marketplace in time for you to take advantage of bargain opportunities.

Have something to sell? A COMPUTER SHOPPER subscriber probably wants to buy it.

Looking for a part, component or even a complete system? A COMPUTER SHOPPER subscriber probably wants to sell it.

COMPUTER SHOPPER is THE marketplace for anything in computers and is read by thousands of people who are ready to buy.

COMPUTER SHOPPER offers a unique format in which classified ads are categorized for fast location of specific items. Combining this with low individual ad rates — 10 cents a word — makes it the ideal place for buyers and sellers to communicate. And, its mix of individual, dealer, and manufacturer ads enable subscribers to find what they want at the best price possible.

COMPUTER SHOPPER will work for you in other ways, too. If you are just thinking about getting into computers, it can help you learn product availability and prices before you make a decision. And, through the timely ads, COMPUTER SHOPPER will keep you abreast of changes in the market which could create bargain opportunities for you.

BUT COMPUTER SHOPPER cannot work for you unless you subscribe.

Want to look us over first? We'll give you your first issue FREE and then bill you for the next 12. If you are not convinced COMPUTER SHOPPER suits your needs, just write "cancel" on the invoice and return it.

And, to let COMPUTER SHOPPER start working for you right now, with a paid subscription we'll also give you a FREE classified ad to sell your pre-owned equipment or to find equipment you want.

If you don't need to use the free classified ad now, use it anytime during your subscription.

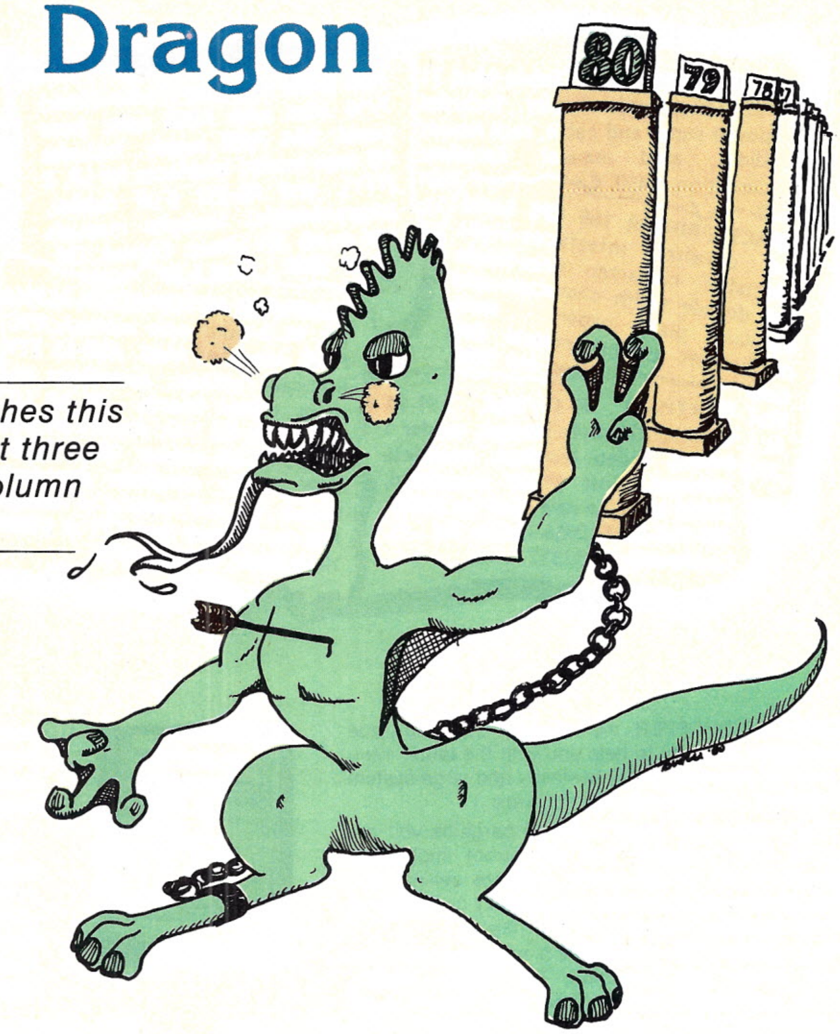
Subscription: $10/year, 12 issues plus your first free one. Bank cards accepted. Money back guarantee.

The Nationwide Marketplace for Computer Equipment

COMPUTER SHOPPER ✓36

P.O. Box F20 • Titusville, Florida 32780
Telephone 305-269-3211

Slaying the 80-Column Dragon



The author/knight-errant searches this fair land and finds not one, but three honest manufacturers of 80-column boards for the Apple II.

Michael S. Tomczyk
418 Arguello Blvd., Suite 2
San Francisco, CA 94118

It's been said for a long time that whoever came up with an 80-column board for the Apple II would be a hero. During the fast few months, several groups of heroes have slain the 80-column dragon, and the implications for Apple II users are enormous.

There are three boards currently available, and more are on the horizon. First on the scene was Sup'R'Terminal from M&R Enterprises, followed closely by Doublevision from Computer Stop and Videoterm from Videx.

After using all three boards and meeting the various dragon-slayers, I still don't know which board is best. All three boards are more or less comparably priced ($295–$395), convert the Apple II display to 80 columns upper/lowercase, work with Pascal and the Apple BASICs and have different drawbacks, advantages and trade-offs.

Before the advent of these boards, you were confined to 40-column, uppercase displays on your Apple video monitor. But the standard typewriter page is 80 columns across, so how could you do word processing? How could you use Pascal, which is formatted for 80 columns? How could you access advanced time-sharing programs, all geared to 80 columns?

It was possible to do these functions, but with great difficulty. For example, you could print out 80 columns of upper/lowercase information, but you could only view 40 columns on your monitor. The new 80-column boards solve the problem, each in a slightly different way. To help sort out the differences between the boards, I've included a chart (see Table 1) showing their comparative features.

One of the best features of these plug-in boards is their compatibility with Pascal. You can now take full advantage of this versatile high-level language in the 80-column format it was originally intended for.

However, you have to use a video monitor—not a television set—to display the characters because with 80 columns you have to use smaller characters, which don't display well on a coarse-resolution television screen. If you use your Apple with a television set, you have to buy a separate black and white video monitor to display 80 columns. Also, you can't use any of the 80-column boards with Apple's hi-res graphics or color.

Sup'R'Terminal

The largest of the three boards, the Sup'R'Terminal from M&R Enterprises, includes the most firmware and the most special features. The character set is excellent, with true descenders, and there are two ingenious adjustments *on the board* which fine-tune the monitor display. The video balance circuit (VBC) tones down the horizontal portions of individual letters (such as the top of the letter T) that are normally displayed much brighter than the ver-

Features	Sup'R'Terminal (M&R Enterprises)	Doublevision (Computer Stop)	Videoterm (Videx)
Price	$395 (complete)	$295 (incl. BASIC & PASCAL disks)	$345 (board) $12 (switchplate assembly)
Slot number	All three boards are slot-independent, but slot #3 is recommended, especially with Pascal		
Display quality	Excellent—two board-based adjustments	Good	Display spills off screen, requires monitor adjustments
Switching (between Apple monitor and board display)	Manual (transfer video jack or use two monitors)	Manual (transfer jack or use two monitors)	Switchplate assembly available from Videx
Compatible with	Pascal, Apple BASICs	Pascal (with disk) Apple BASICs (with disk)	Pascal, Apple BASICs
Monitor	All three boards require video monitor instead of television set, although a crude display is possible using a television set with rf modulator		
Synchronous/asynchronous	Synchronous	Asynchronous	Asynchronous
Light pen	No (possibly in future)	Light pen connection (pen being developed)	Videx light pen connection and pen available
Power	Draws from both +5 V and +12 V power supplies	Draws from +5 V supply only	Draws from +5 V, −5 V and +12 V power supplies
Minimum configuration	Apple with 16K RAM	Apple with 32K RAM and disk drive	Apple with 16K RAM
Firmware	2K board-based	RAM-based software transparent to user	1K board-based
Phone modem	D.C. Hayes compatible	D.C. Hayes compatible	Testing
Character set	128 character ASCII, 5 × 8 dot matrix, true descenders	128 character ASCII, 5 × 7 dot matrix, no true descenders	128 character ASCII, variable dot matrix, true descenders
Documentation	Excellent manual with helpful subroutines for special functions	Comprehensive user-oriented manual	Large spiral-bound manual including schematics and most thorough technical information
User-defined characters	Complete user-defined character sets, immediate access of up to ten fonts, easy instructions	Must change fonts or alphabets by replacing character generator chip	User-defined character (EPROM programmer needed or buy font EPROMs from Videx)
Graphics	Text and semi-graphics. Apple graphics and color must be displayed on separate screen, inhibits Apple color when used simultaneously	Text only. Apple graphics and color fully supported on separate screen	Pre-defined ROM-based graphics character set included, also user-defined EPROM graphics
Inverse mode	Yes, but inverse character set must be defined	Normal inverse	Resoldering required, permanent inverse only
Shift (getting caps from lowercase mode)	CTRL A = single letter shift CTRL A CTRL A = shift lock (all caps) One wire shift modification converts shift key (how-to in manual) CTRL V will toggle modification off/on	ESC = single letter shift ESC ESC = shift lock (must exit lowercase to use normal ESC functions such as cursor moves) One wire shift modification converts shift key, CTRL E locks shift with modification	CTRL A = shift lock CTRL A = shift release (single letter capitalization not automatic, must enter and exit shift mode). New product modifies shift key to work normally
Cursor modes	Eight different cursor modes; fast/slow blink, underline, box, invisible cursor	User-definable cursor modes, up to 20+ combinations	User-definable cursor modes, up to 20+ combinations
Cursor movement	ESC ABCD/ESC IJKM	ESC ABCD/ESC IJKM but must exit to all caps mode (type CTRL E) to use ESC cursor controls	ESC ABCD/ESC IJKM
Tabs	No comma tab, requires special HTAB commands or POKE 36,n, where n = column number or GOTO XY CTRL I = 8 space tabbing	HTAB, comma tab problems (being solved); GOTO XY; POKE	No comma tab in INT BASIC, full comma tabs in Pascal; CTRL shiftN XY positions cursor
Screen display window	"Window Maker" commands define variable window display	Location-POKEs required	Location-POKEs required
Stoplist (interrupts scrolling)	CTRL-S	CTRL-S	---
Left bracket access	CTRL-K accesses left bracket	No	CTRL-K access
Warranty	90 days	90 days	90 days

Table 1. Comparative features of the 80-column boards for the Apple II.

tical part of the letter, especially on less expensive monitors. The VBC adjustment, coupled with an intensity dial and the monitor's own adjustments, produces a superior screen display, even on inexpensive monitors.

Sup'R'Terminal also has the most board-based RAM (3K), which permits user-defined character sets. You can define your own type fonts, scientific or mathematical notation sets or foreign alphabets—up to ten different character sets—and access them immediately. Excellent documentation shows clearly how to do this.

The only drawback is the lack of a full user-defined graphics feature. Because there is a one-dot gap between each character's dot matrix block, you can't, for instance, define a totally smooth graphic image. You have to settle for "semi-graphics."

Another convenient feature is the "window maker," which lets you easily define a window of any size or shape on the screen, with everything displayed inside that window. You could put a "menu" or other reference system on one part of the screen, then use the scrolling window to run your programs. Normally you need a complex software package to do this.

Sup'R'Terminal is also the only board that is synchronous with Apple II. It uses Apple's internal timing system, so the board should run slightly faster than asynchronous boards.

I've heard several people claim that Sup'R'Terminal draws too much power and might overtax Apple's power supply. My own tests show that Apple's own plug-in boards draw more power from the 5 V power supply provided than Sup'R'Terminal.

Apple II has a power problem that is not tied to any one board or group of boards. You run the risk of overtaxing the power supply if the total power drain of all the boards being used exceeds the specifications set down by Apple (see page 92 of the reference manual).

There are several combinations of plug-in boards and peripherals that can cause this problem, but, unfortunately, not all boards reveal how much power they use, so it's hard to determine when you're approaching the limit. The best interim solution is to unplug any boards you absolutely don't need when all the slots are filled in a fully configured system.

Doublevision

Doublevision from Computer Stop differs

from the other terminals in that it is software-based and uses disks to provide its features. The disk can be easily updated, or new features added (by Doublevision). You have to boot up the disk each time you turn on the board, and there exists the danger of damaging one of the disks and the possibility that eventually you'll use an advanced machine-level program that "clobbers" some RAM memory locations, although the chance of this happening is remote.

There are two disks included with the system—one for BASIC and one for Pascal. When using Pascal, the BASIC disk must be booted up first.

Doublevision's character set doesn't have true descenders, but the descenders print out normally on printers having true descender character sets. Incidentally, all three boards work well with thermal printers. I tested them with the Trendcom 200 and Apple's new Silentype, with good results.

Doublevision requires the use of the ESC key as a shift, which makes for fast typing. It is probably the fastest shift key to use, because you only have to hit one key; however, you have to exit the lowercase mode if you want to use any of the normal ESC functions, such as cursor movement.

Doublevision is the least expensive of the three boards ($295) and provides excellent 80-column power at the lower end of the kilobuck range.

Videoterm

Like Doublevision, Videoterm by Videx created a lot of excitement when it was introduced at the Computer Faire. It offers the most technically well-documented manual of all the boards, although it seems a trifle weak in the explanation and use of board functions.

Because of the unique full-screen character set format, the display tends to spill off the screen around the edges and requires you to adjust the horizontal and vertical screen size controls, which are located *inside* most monitors. Videx claims their board works best with the Leedex monitor.

Videoterm also has an inexpensive ($12) switchplate, which enables you to access either the Apple 40-column display or the Videoterm 80-column display on the same monitor simply by flicking a switch. For example, you can go back to the Apple display system for graphics without having to unplug the monitor from the board and plug it back into the Apple, which you have to do with Sup'R'Terminal and Doublevision.

By using a character set EPROM, you can define your own character sets or graphics, although you need an EPROM programmer to do this. Videx conveniently sells a variety of predefined EPROMs that you can use to change fonts simply by substituting the EPROM chip.

Videoterm also has a pre-defined graphics set built into the system. The set essentially lets you do the line-drawing and is especially helpful if you want to create business forms or graphs. The documentation showing how to use this feature is clever.

The major drawback of the Videx board is its awkward shift command. The other boards provide a single-letter shift key that automatically shifts back down after execution. Every time you capitalize a single letter with Videoterm, you must type a command to get into the uppercase mode, then type another command to go to lowercase. However, Videx is already solving this problem with a new product: a PC board you substitute for one of the IC chips to convert the "old" Apple keyboard to the "new" keyboard. It turns the shift key into a normal shift key and provides a CTRL RESET function, with no special hardware modification. It can be used to convert any Apple.

With the other boards, you need a one-wire modification to convert the shift key to

a normal shift key used in ordinary typing. Sup'R'Terminal shows you how to make this modification in their manual, although this mod will void your Apple warranty. I suggest making this modification if you buy a Sup'R'Terminal or Doublevision, especially if you do a lot of word processing.

On the Horizon

Another 80-column Apple board I did not include in Table 1 was developed by Chuck Mauro, a 21-year-old Apple computer engineer who developed the board in his spare time and has recently started production.

The board uses Synertek's new 6545 chip, which provides a little extra capacity and allows for some extra features, such as medium resolution (160 × 72 lines) vector graphics, which may be used simultaneously with text (in black and white). This asynchronous board provides normal functioning tabs and has an inverse mode and keyboard-controlled monitor switching (like Videoterm's switchplate, only you control from the keyboard).

Chuck's board, tentatively priced at $360, will be sold through Apple dealerships under the corporate name "Sum Apple Software."

Outlook for 80-Column Boards

It's obvious that the makers of these boards will be successful, because most Apple users want 80 columns and the special features described above. However, the market for these boards is not infinite. Apple III will definitely provide an 80-column format. Thus, while Apple IIs may continue to be sold, the market for 80-column boards will start falling off next year.

This means that the entrepreneur-engineer-inventors who are lauded today as mighty dragon-slayers will have to slay more dragons early next year to keep going. Most of the board designers I talked to are already working on new products, so keep an eye on these companies to see what happens. ■

MOM'S

MAIL ORDER MICROS ✓53

Dept 8KB • PO Box 427 • Marlboro NH 03455
Phone: (603) 924-3041*

SALE

5% EXTRA DISCOUNT
for orders with checks or money orders (see "TERMS" for details).
ITEMS THIS PAGE ONLY
ORDER NOW—
Prices subject to change without notice.

SCOTT ADAMS ADVENTURES

SPECIAL 10% OFF REGULAR PRICES
(Reg $14.95)

#1 ADVENTURELAND

YOU'LL WANDER THROUGH AN ENCHANTED LAND ENCOUNTERING WILD ANIMALS AND MAGICAL BEINGS WHILE YOU TRY TO RECOVER LOST TREASURES.
Order #SA001T for TRS Level II 16K, #SA001S for Sorcerer 16K, #SA001A for Apple 24K—$13.45 each, on cassette.

#2 PIRATE ADVENTURE

SAIL TO TREASURE ISLAND AND TRY TO RECOVER LONG JOHN SILVER'S LOST TREASURES.
Order #SA002T for TRS Level II 16K, #SA002S for Sorcerer 16K, #SA002A for Apple 24K—$13.45 each, on cassette.

#3 MISSION IMPOSSIBLE ADVENTURE

SAVE THE WORLD'S FIRST AUTOMATED NUCLEAR REACTOR WHEN YOU COMPLETE YOUR MISSION.
Order #SA003T for TRS Level II 16K, #SA003S for Sorcerer 16K, #SA003A for Apple 24K—$13.45 each, on cassette.

#4 VOODOO CASTLE

SAVE COUNT CRISTO FROM THE FIENDISH CURSE OR FOREVER BE DOOMED—BEWARE THE VOODOO MAN.
Order #SA004T for TRS Level II 16K, #SA004S for Sorcerer 16K, #SA004A for Apple 24K—$13.45 each, on cassette.

#5 THE COUNT

LOVE AT FIRST BYTE FROM YOUR BIG BRASS BED IN TRANSYLVANIA.
Order #SA005T for TRS Level II 16K, #SA005S for Sorcerer 16K, #SA005A for Apple 24K—$13.45 each, on cassette.

#6 STRANGE ODYSSEY

YOU'RE MAROONED AT THE GALAXY'S EDGE AND DISCOVER RUINS OF AN ANCIENT ALIEN CIVILIZATION AND TRY TO COPE WITH UNEARTHLY TECHNOLOGIES WHILE YOU AMASS FABULOUS TREASURES.
Order #SA006T for TRS Level II 16K, #SA006S for Sorcerer 16K, #SA006A for Apple 24K—$13.45 each, on cassette.

#7 MYSTERY FUN HOUSE

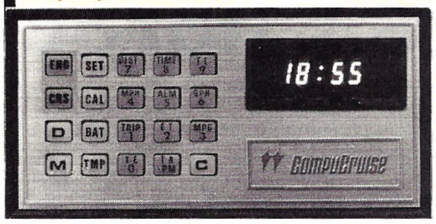
FIND YOUR WAY THROUGH THE STRANGEST FUN HOUSE BEFORE THE WEIRD PARK CLOSES.
Order #SA007T for TRS Level II 16K, #SA007S for Sorcerer 16K, SA007A for Apple 24K—$13.45 each, on cassette.

#8 PYRAMID OF DOOM

AN EGYPTIAN TREASURE HUNT THROUGH A NEWLY UNCOVERED PYRAMID, COMPLETE WITH ANCIENT CURSE.
Order #SA008T for TRS Level II 16K, #SA008S for Sorcerer 16K, #SA008A for Apple 24K—$13.45 each, on cassette.

If you enjoy driving, you're going to get a COMPUCRUISE. Once you see what it can do, you just won't be able to live without it.

COMPU CRUISE



MODEL # 41

$127.95

This gadget fits into most dashboards . . . no strain even in a tiny sports car like the Mazda RX-7 . . . and once you have it, every trip is like flying a 747. The darned thing tells you the time, how fast you're going, how far you've been on this trip or since the last regassing, how many miles per gallon you're getting, either at the instant or the average on the trip . . . or gallons per hour at the moment or for the trip . . . temperature outside . . . inside (or coolant temperature, if you prefer) . . . oh, it has an elapsed time for the trip, a stop watch, lap time, an alarm . . . how much further for your trip, how many gallons more the trip will take, how much longer for the trip at your present average speed . . . yes, it gives you your average speed for the trip. You prefer it in metric, no strain . . . liters remaining, etc. Did we mention that it also has cruise control either at a speed set on the control board or at whatever speed you are traveling? The Compucruise will keep you busy and entertained during any trip . . . telling you more than you will ever want to know.

The Compucruise is not difficult to install . . . though it does connect to everything except the cigarette lighter. Until you've tried computerized travel, you haven't found out how much fun driving can be. It will work on any car not having fuel injection . . . and there is a front-wheel drive accessory gadget available for only $4.40-#P001 (regularly $5.50).

The price for the Compucruise is regularly $199.95 . . . and a bargain at that price. We'll sell you one of these fantastic gadgets for $159.95 with cruise control (Model 44-#P002), and $127.95 without (Model 41-#P003).Send money . . . and start having fun!

CASIO'S NEW C-80 CALCULATOR CHRONOGRAPH

has lightweight attractive ruggedly built black plastic water-tight case and band, regular digital watch features of hours, minutes, seconds, AM/PM and day on display PLUS two time zones, calendar and a 4 function calculator! You've seen it advertised in the Wall Street Journal for $75 . . . MOM's price is only $49.95. Hurry our supply is limited . . . order #PC80 now.

WAMECO S-100 Bare Computer Boards

			Retail	MOM'S Price	
10 ea	CPU-1	8080 CPU Board	26.95	22.95	#LP001
10 ea	MEM-1A	8K Memory Board (2102)	25.95	21.95	#LP002
10 ea	Mem-2	16K Memory Board (2114)	26.95	22.95	#LP003
10 ea	EPM-2	2708/2716 EPROM Board	25.95	21.95	#LP004
10 ea	EPM-1	1702 EPROM Board	25.95	21.95	#LP005
10 ea	RTC-1	Real Time Clock	23.95	20.50	#LP006
10 ea	QM-9	The Little Mother (9 slot)	30.95	26.50	#LP007
10 ea	QM-12	The Quiet Mother (12 slot)	34.95	29.95	#LP008
5 ea	FDC-1	Floppy Disk Board	42.95	36.95	#LP009
7 ea	FPB-1	Front Panel IMSAI size)	44.50	37.95	#LP010
1 ea	EPM-1	1702 EPROM Board (asmbld) without EPROMS	49.95	42.50	#LP011

BOOK CLEARANCE

50% OFF OR MORE!

How To Buy And Use Minis And Micros (Sams-publisher) #BK 1025—was $9.95 now $5.
How To Program Microcomputers (Sams-Publisher) #BK1027—was $9.95 now $4.50.
8080A Bugbook (Sams-publisher) #BK1103—was $9.95 now $5.
Understanding Microcomputers (Scelbi-publisher) #BK1079—was $8.30 now $4.
Your Own Computer (Sams-publisher) #BK1072—was $1.95 now $1.
Underground Buying Guide (PMS-publisher) #BK1067 was $5.45 now $2.50.
Periodical Guide 1976 (Berg-publisher) #BK1041—was $3 now $1.50.
Periodical Guide 1977 (Berg-publisher) #BK1042—was $3 now $1.50.
Compulator Book (Tab-publisher) #BK1012—was $7.95 now $4.

ASK MOM

For details of October 1980 Asian and European Electronics Tours. See January Wayne Green editorials in Kilobaud Microcomputing and 73 Magazine. Info #002 . . . No charge.

MAGIC LIGHT BULB SAVER

If you've got a light bulb you need our MAGIC L.B.S. which increases the life expectancy of a new bulb 93 times! . . . and saves you 44% on energy costs. Regularly $2.50 . . . MOM's price is $1.99 each. #LS001.

TDL SOFTWARE-DISK

TDL FDOS & SuperBASIC on 8" Disk—This requires an ICOM Disk Controller and at least 20K of memory, plus a ZAPPLE Monitor in an S-100 Cabinet (Altair, IMSAI, etc.) #D0065SX-$120.

TDL System Software on 5¼" disk—This set of system software requires a North Star Disk Controller, and a TDL Systems Monitor Board I, and consists of 12K BASIC, Relocator/Linking Loader, Z-80 Editor, and Text Processor. #D0066X-$160.

TDL System Software on 5¼" disk—This is the same as above, but does not require the Systems Monitor Board I. #D0067X-$180.

TDL System Software on 5¼" disk—Again, as above, but requires a HELIOS Disk Controller and the TDL Systems Monitor Board II (not I). #D0068X-$160.

VECTOR GRAPHIC S-100

Vector Graphic ROM/RAM card—12K empty ROM sockets, 1K RAM, excellent condition. No documentation. (Make offer—we may surprise you and accept!) #D0078X-$95.

HONEYWELL

Honeywell ASR-33 Communications Consoles with TTY, paper tape reader and punch. Used, working when removed from service. Shipped freight collect or you pick up. Weight 300 lbs. $325. Order #P006X.

PANASONIC TAPE DECKS

Panasonic RS261 US Stereo Cassette Decks—with auto-stop, record level adjust, VU meters, used condition; all have had heads replaced and aligned. #T001-$50.

HEAD ALIGNMENT KIT

Best cassette recorder tape head alignment kit available. Solves loading problems. #K001-only $9.95.

COMPUCOLOR HARDWARE

Compucolor Computer 8001 (use as computer or 75 MHZ Color Monitor)—8K RAM, BASIC and DOS in ROM, good condition. #S025X-$1200 each. **SPECIAL PRICE.**

Compucolor MiniFloppy—5¼ inch, good condition. #S026X-$475 each. **SPECIAL PRICE.**

Compucolor 8K RAM card—Static RAM, good condition. S027X-$180 each. **SPECIAL PRICE.**

Compucolor Floppy Tape Drive—Uses eight-track cartridges, good condition. #S028X-$55 each. **SPECIAL PRICE.**

PROCESSOR TECH SOFTWARE

Processor Tech Extended Disk BASIC—This is full disk BASIC on 8" disk for HELIOS II disk controllers with PTDOS and greater than 16K. #D015S-$70 each.

BASIC 5 from Processor Tech—This is a simple BASIC for a SOLOS, CUTTER, or CONSOL Monitor and 8K of RAM. #D013-$11.60 each.

Processor Tech GAMEPAC for above BASIC—Various simple games. #D014-$10.00 each

TERMS: EXTRA 5% DISCOUNT ALLOWED ONLY ON ORDERS ACCOMPANIED BY CASHIERS CHECK, PERSONAL CHECK OR MONEY ORDER. NO C.O.D.'S PLEASE. ADD $2.50 FOR HANDLING FOR ORDERS UP TO $50.00—ALL OTHERS ADD 5% (WE REFUND EXCESS). WE SHIP UPS OR INSURED MAIL IF POSSIBLE, WE ACCEPT MC, VISA OR AMEX.

PROCESSOR TECH SOL

SOL Computers—8K RAM Monitor, S-100, excellent condition. #D004X-$850.00 each.

TREK-80 on cassette for SOL—This is one of the best real time space games available today; needs 8K. #D005X-$9 each.

Graphics Character Generator

You can mix text and graphics anywhere on your Apple II screen with this CHAR-GRAF program.

Robin B. Moore
Warner Hill Rd., RFD #5
Derry, NH 03038

One of the nicest features of the Apple II computer is its high-resolution bit-mapped graphics display mode. It can be used for graphs, plots, digitized photographs or colorful, arcade-quality video games. Its only significant limitation is its inability to fully mix text and graphics.

Normally, text is assigned only to the bottom four lines of the screen in the hi-res graphics mode. This is not hard to live with, but it would be nice to mix text and graphics anywhere on the screen. Another improvement is the addition of lowercase and user-definable characters.

These two minor changes would create a machine with video display flexibility unmatched in the current personal computer market. There is nothing to prevent the Apple from displaying text on the hi-res screen. It simply isn't programmed to do so!

The Program

CHAR-GRAF is a package of assembly-language software that will allow you to add the following features to your Apple II:

- Fully mixed text and graphics in hi-res mode.
- Lowercase and uppercase characters.
- User-definable scientific and game characters.
- Multiple resident character sets.
- An invert the hi-res screen function.

These can be added without hardware changes. It will take a little time to type all the code into your Apple, but you will only have to do it once.

I designed CHAR-GRAF to be as compatible with existing Apple software as possible. It resides in lower memory along with the Apple II hi-res graphics routines that are provided by Apple as the first part of the hi-res demo tape. Its operation requires the presence of these routines. (If you are using the routines in the Programmer's Aid #1 ROM, also from Apple, CHAR-GRAF will require only minor changes.)

Printing text onto the hi-res screen with CHAR-GRAF is accomplished using normal PRINT statements. Text is displayed in the same 5 × 7 dot matrix format that is normally used by the Apple and will appear on the screen in its normal positions. VTABs and TABs will work as they normally do, as will user-defined scrolling windows.

In its present implementation, CHAR-GRAF will not scroll the hi-res screen, nor will it exhibit any response to the normally used screen clearing functions,

*Listing 1. Assembler source code for the CHAR-GRAF routines. This listing was created using the Microproducts assembler for the Apple II (which I have since abandoned, for many reasons). Base page symbols are shown preceded by an *. If you have an assembler, you can reassemble CHAR-GRAF to reside anywhere you want in memory.*

```
0010  :* TITLE: CHAR-GRAF  V1.3       *
0020  :*******************************
0030  :*                              *
0040  :* CHAR-GRAF............V1.3    *
0050  :*                              *
0060  :* SOFT CHARACTER GENERATOR     *
0070  :* ROUTINES FOR THE APPLE II.   *
0080  :*                              *
0090  :* LINKS TO BASIC VIA CSWL,H    *
0100  :* FOR AUTO ARGUMENT PASSING.   *
0110  :*                              *
0120  :* COPYRIGHT 3/1/1979  BY       *
0130  :*                              *
0132  :* ROBIN B. MOORE               *
0133  :* WARNER HILL RD. RFD # 5      *
0134  :* DERRY, N.H. 03038            *
0135  :*                              *
0140  :* ALL COMMERCIAL RIGHTS        *
0142  :* RESERVED.                    *
0150  :*                              *
0160  :*******************************
0170  :*
0180  :*
0190  :*   LOCATION EQUATES
0200  :*
0210  :*
0215  DCOL .DL 001C     SHAPEDRAW COLOR
0220  TPL  .DL 001E     TABLE PTR. L
0230  TPH  .DL 001F       ''   ''  H
0240  WNDW .DL 0021     WINDOW WIDTH
0250  CH   .DL 0024     CURSOR HORIZ
0260  GBSL .DL 0026     HIRES BASE ADDL
0270  GBSH .DL 0027       ''   ''  ADDH
0280  BASL .DL 0028     TEXT BASE ADDL.
0290  BASH .DL 0029     TEXT BASE ADDH.
0300  IFLG .DL 0032     INVERSE FLAG
0310  CSWL .DL 0036     CONSOLE PTR. LO
0320  CSWH .DL 0037       ''     '' HI
0322  A1L  .DL 003C     ADD PTR LO
0324  A1H  .DL 003D      ''   HI
0330  XSAV .DL 0045     SAVED X-REG
0340  YSAV .DL 0046      ''   Y-REG
0350  CSAV .DL 0047     SAVED CHAR CODE
0360  SFLG .DL 0048     STRIKEOVER FLAG
0365  HCOL .DL 032C     PLOT/LINE COLOR
0370  PAGE .DL 032E     HIRES PG. 20/40
0380  TABL .DL 0330     TABLE BASE ADDL
0390  TABH .DL 0331     TABLE BASE ADDH
0400  COUT .DL FDF0     APPLE OUTPUT
0410  :*
0420  :*
0430  :* LINK IN AND INITIALIZE
0440  :*  IN BASIC, CALL 2048
0450  :*
0460       OR 0800
0470  LINK PHA
0480       LDA 08       SETUP CSWL,H
0490       STA *CSWH    TO LINK THRU
0500       LDA OF       CGRAF ROUTINES
0510       STA *CSWL
0520       LDA OFF      DISABLE STRIKE-
```

which must be accomplished manually through the use of TABs, VTABs and the hi-res screen clear functions. Limited areas may also be cleared by simply moving the cursor to the desired position and printing blanks.

These limitations are unfortunate, but after my experience writing the demo program and the Character Set Editor, I've found that they are not difficult to deal with. It is also easy to simulate the Apple's normal flashing cursor in hi-res mode.

In part payment for the missing scroll and screen clear functions, I've added a few new ones. With a CALL to INVSCRN (2187), you can inverse-video the entire hi-res screen and also complement the current text mode and hi-res color. A CALL to STRIKEOVER (2180) will set CHAR-GRAF into strikeover mode, causing its output to be ORed to the screen instead of overwriting as it normally would.

Also, I have added a pseudo-CHR$ function. If you POKE an ASCII character code + 128 into CHARLOC (71) and then CALL CHARDRAW (2172), the equivalent character will be printed at the current cursor position.

How CHAR-GRAF Works

Whenever text is output by the Apple, control is passed to the output routine whose address is stored in two base-page locations called CSWL and CSWH ($36,37). Normally, this is the video output routine in the monitor ROMs. When CHAR-GRAF is in use, these pointers are changed to point to the entry address of its text output section. This is done easily from BASIC with a CALL LINK (2048).

The CSWL, CSWH pointers may be reset to their normal values with a PR#0 command, which will turn off CHAR-GRAF. You will need to do a TEXT command to take the Apple out of hi-res mode if you wish to see any further text output.

To allow CHAR-GRAF to use a number of different character sets at once, there is another pointer set, TL and TH (816, 817). These point to the beginning of the character set currently in use.

Changing character sets is done by changing these pointer

```
0530          STA *SFLG    OVER MODE.
0540          PLA
0550    LRTS  RTS          RETURN
0560    :*
0570    :*
0580    :*  CHARACTER GENERATOR
0590    :*  ROUTINES FOLLOW.
0600    :*
0610    :*  ENTRY AT CHAR ( $80B )
0620    :*  SAVES ALL REGISTERS
0630    :*
0640    :*
0650    CHAR  STX *XSAV    SAVE X REG
0660          STY *YSAV      ''  Y
0670    C1    PHA             ''  A
0680    C2    LDX 00       CLR X-REG
0690    CR?   CMP 8D       CR ?
0700          BNE LLIM     NO
0705          LDA 0A0      GET A SPACE
0710    CR1   LDY CH       GET CURSOR HOR.
0715          INY
0730          CPY *WNDW    >= WNDW-1 YET?
0740          BEQ C2       YES, DONE
0760          JSR C1       CALL YOURSELF
0770          JMP CR1
0780    LLIM  SEC          SETUP FOR SBC
0790          SBC 0A0      CHAR < $A0?
0800          BCC EXIT     YES, EXIT
0810    :*
0820    :*
0830    :*  PLOTTING ROUTINE
0840    :*
0850    :*  PLOTS CHAR IN HIRES AT POSN
0860    :*  EQUIV TO CURRENT CH,CV.
0870    :*
0880    :*
0890    PLOT  STX *TPH     CLEAR CHAR PTR.
0900          ASL
0910          ROL *TPH
0920          ASL
0930          ROL *TPH
0940          ASL          TPL,TPH =
0950          ROL *TPH     A.C.  * 8
0960          CLC               +
0970          ADC TABL     TABLE START
0980          STA TPL      LOCATION
0990          LDA TPH
1000          ADC TABH
1010          STA *TPH
1020          LDA *BASL    TEXT BASE ADD.
1030          ADC *CH      ADD CURSOR HPOS
1040          STA *GBSL    HIRES BASE LO
1050          LDY 07       SETUP Y WITH $7
1060    SCAN  LDA *BASH    ABCDEFGH ->
1070          AND 03       00000FGH ->
1080          STA *GBSH       GBSH
1090          TYA          GET STROKE CNT.
1100          ASL
1110          ASL          A.C.  = Y*4
1120          ORA *GBSH
1130          ORA PAGE     'OR' WITH 20/40
1140          STA *GBSH    GBSL,H ALL SET
1150          BIT *IFLG    TEST INV MODE
1160          LDA (TPL ),Y GET CHAR SLICE
1170          BVS NONI     NOT INV MODE
1180          EOR OFF      INVERT SLICE
1190    NONI  BIT *SFLG    'OR' TO SCRN?
1200          BVS NORM
1210          ORA (GBSL,X) 'OR' SCRN W/DAT
1220    NORM  STA (GBSL,X) WRITE TO SCRN
1230          DEY          NEXT SLICE?
1240          BPL SCAN     YES, LOOP AGAIN
1250    :*
1260    :*
1270    :*  EXIT SECTION
1280    :*
1290    :*
1300    EXIT  LDX *XSAV    RESTORE X
1310          LDY *YSAV      ''  Y
1320          PLA            ''  A
1330    CRTS  JMP COUT     GOODBYE!
1340    :*
1350    :*
1360    :*  ROUTINE USED BY BASIC TO
1370    :*  PRINT CHAR STORED IN CSAV
1380    :*  (FROM BASIC, CALL 2172)
1390    :*
1400    CDRW  PHA          SAVE A-REG
1410          LDA *CSAV    GET CHAR CODE
1420          JSR CHAR     DISPLAY IT
1430          PLA          GET BACK A-REG
1440          RTS
1450    :*
1460    :*
1470    :* CALL HERE TO SET STRIKEOVER
1480    :* MODE. (FROM BASIC, CALL 2180)
1490    :*
1500    :*
1510    SMOD  PHA
1520          LDA 00       ENABLE
1530          STA *SFLG    STRIKEOVER FLG.
1540          PLA
1550          RTS
1570    :*
1580    :*
1590    :* CALL HERE TO INVERT ENTIRE
1600    :* HI-RES SCREEN, COLOR, AND
1610    :* TEXT MODE.
1620    :*   (FROM BASIC, CALL 2187)
1630    :*
1640    :*
1650    INV   PHA
1660          STY *YSAV
1670          STX *XSAV    SAVE A,X,Y REGS
1680          LDA PAGE     CURRENT PAGE
1690          STA *A1H
1700          LDX OFF
1710          LDY 00
1720          STY *A1L     PTRS SET UP
1730    INV1  TXA          $FF -> AC
1740          EOR (*A1L),Y GET COMP DATA
1750          STA (*A1L),Y ONTO SCREEN
1760          INY          NEXT LOC
1770          BNE INV1
1780          INC *A1H
1790          LDA *A1H
1800          AND 1F       MASK OFF 20/40
1810          BNE INV1     NEXT PAGE
1820          LDA *IFLG
1830          EOR OC0      COMP MSB'S
1840          STA *IFLG    SET OPP. MODE
1850          LDA HCOL     GET HCOLOR
1860          EOR OFF      COMPLEMENT IT
1870          STA HCOL     REPLACE IT
1872          LDA *DCOL    GET DRAW COLOR
1873          EOR OFF      COMPLEMENT IT
1874          STA *DCOL    RESTORE IT
1880          LDX *XSAV
1890          LDY *YSAV
1900          PLA          RESTORE A,Y,X
1910          RTS
1920    ENDD   EN
```

```
800LLLLLL
0800-   48          PHA
0801-   A9 08       LDA   #$08
0803-   85 37       STA   $37
0805-   A9 0F       LDA   #$0F
0807-   85 36       STA   $36
0809-   A9 FF       LDA   #$FF
080B-   85 48       STA   $48
080D-   68          PLA
080E-   60          RTS
080F-   86 45       STX   $45
0811-   84 46       STY   $46
0813-   48          PHA
0814-   A2 00       LDX   #$00
0816-   C9 8D       CMP   #$8D
0818-   D0 10       BNE   $082A
081A-   A9 A0       LDA   #$A0
081C-   AC 24 00    LDY   $0024
081F-   C8          INY
0820-   C4 21       CPY   $21
0822-   F0 F0       BEQ   $0814
0824-   20 13 08    JSR   $0813
0827-   4C 1C 08    JMP   $081C
082A-   38          SEC
082B-   E9 A0       SBC   #$A0
082D-   90 45       BCC   $0874
082F-   86 1F       STX   $1F
0831-   0A          ASL
0832-   26 1F       ROL   $1F
0834-   0A          ASL
0835-   26 1F       ROL   $1F
0837-   0A          ASL
0838-   26 1F       ROL   $1F
083A-   18          CLC
083B-   6D 30 03    ADC   $0330
083E-   8D 1E 00    STA   $001E
0841-   AD 1F 00    LDA   $001F
0844-   6D 31 03    ADC   $0331
0847-   85 1F       STA   $1F
0849-   A5 28       LDA   $28
084B-   65 24       ADC   $24
084D-   85 26       STA   $26
084F-   A0 07       LDY   #$07
0851-   A5 29       LDA   $29
0853-   29 03       AND   #$03
0855-   85 27       STA   $27
0857-   98          TYA
0858-   0A          ASL
0859-   0A          ASL
085A-   05 27       ORA   $27
085C-   0D 2E 03    ORA   $032E
085F-   85 27       STA   $27
0861-   24 32       BIT   $32
0863-   B1 1E       LDA   ($1E),Y
0865-   70 02       BVS   $0869
0867-   49 FF       EOR   #$FF
0869-   24 48       BIT   $48
086B-   70 02       BVS   $086F
086D-   01 26       ORA   ($26,X)
086F-   81 26       STA   ($26,X)
0871-   88          DEY
0872-   10 DD       BPL   $0851
0874-   A6 45       LDX   $45
0876-   A4 46       LDY   $46
0878-   68          PLA
0879-   4C F0 FD    JMP   $FDF0
087C-   48          PHA
087D-   A5 47       LDA   $47
087F-   20 0F 08    JSR   $080F
0882-   68          PLA
0883-   60          RTS
0884-   48          PHA
0885-   A9 00       LDA   #$00
0887-   85 48       STA   $48
0889-   68          PLA
088A-   60          RTS
088B-   48          PHA
088C-   84 46       STY   $46
088E-   86 45       STX   $45
0890-   AD 2E 03    LDA   $032E
0893-   85 3D       STA   $3D
0895-   A2 FF       LDX   #$FF
0897-   A0 00       LDY   #$00
0899-   84 3C       STY   $3C
089B-   8A          TXA
089C-   51 3C       EOR   ($3C),Y
089E-   91 3C       STA   ($3C),Y
08A0-   C8          INY
08A1-   D0 F8       BNE   $089B
08A3-   E6 3D       INC   $3D
08A5-   A5 3D       LDA   $3D
08A7-   29 1F       AND   #$1F
08A9-   D0 F0       BNE   $089B
08AB-   A5 32       LDA   $32
08AD-   49 C0       EOR   #$C0
08AF-   85 32       STA   $32
08B1-   AD 2C 03    LDA   $032C
08B4-   49 FF       EOR   #$FF
08B6-   8D 2C 03    STA   $032C
08B9-   A5 1C       LDA   $1C
08BB-   49 FF       EOR   #$FF
08BD-   85 1C       STA   $1C
08BF-   A6 45       LDX   $45
08C1-   A4 46       LDY   $46
08C3-   68          PLA
08C4-   60          RTS
08C5-   C0 C0       CPY   #$C0
08C7-   C0 C0       CPY   #$C0
08C9-   C0 C0       CPY   #$C0
08CB-   C0 C0       CPY   #$C0
08CD-   C0 C0       CPY   #$C0
```

Listing 2. CHAR-GRAF routines from the Apple's built-in disassembler. CHAR-GRAF may be entered in either this form using the Apple's mini-assembler, or the hex codes only entered using the monitor.

values. For example, if you wanted to use a character set that was located starting at 1234, you would set TL and TH with the following commands:

```
POKE TL,1234 MOD 256 : POKE TH,1234 / 256
```

When CHAR-GRAF receives a character for output, it saves the current 6502 register contents and then does the following:

1. If the character is a return code, blank the rest of the current line and go to step 4.

2. If the character is any other control code, go to step 4.

3. For any other character, find the character set address in TL,TH, and use it to compute the address of the current character within the set. Use the current cursor horizontal and vertical positions to compute the equivalent hi-res screen address. Draw the character on the hi-res screen in the current video mode (normal or inverse). Go to step 4.

4. Restore all the 6502 registers to their entry states and jump to the normal video output routine. This causes the cursor position to be updated normally for each character output.

The assembler source code for CHAR-GRAF is shown in Listing 1. The disassembled object code is shown in Listing 2.

Digital IC Probe & Logic Pulser

PRB-1 DIGITAL LOGIC PROBE

Compatible with DTL, TTL CMOS, MOS and Microprocessors using a 4 to 15V power supply. Thresholds automatically programmed. Automatic resetting memory. No adjustment required. Visual indication of logic levels, using LED's to show high, low, bad level or open circuit logic and pulses. Highly sophisticated, shirt pocket portable (protective tip cap and removable coil cord).

Automatic threshold resetting • DE to > 50 MHZ
Compatible with all logic families 4-15 VDC • 10 Nsec. pulse response
Supply O.V.P. to ± 70 VDC • 120 K Ω impedance
No switches/no calibration • Automatic pulse stretching to 50 Msec.
Open circuit detection • Automatic resetting memory
Range extended to 15-25 VDC with optional PA-1 adapter

PLS-1 LOGIC PULSER

The PLS-1 logic pulser will superimpose a dynamic pulse train (20 pps) or a single pulse onto the circuit node under test. There is no need to unsolder pins or cut printed-circuit traces even when these nodes are being clamped by digital outputs.

PLS-1 is a multi-mode, high current pulse generator packaged in a hand-held shirt pocket portable instrument. It can source or sink sufficient current to force saturated output transistors in digital circuits into the opposite logic state.

Signal injection is by means of a pushbutton switch near the probe tip. When the button is depressed, a single high-going or low-going pulse of 2μ sec wide is delivered to the circuit node under test. Pulse polarity is automatic: high nodes are pulsed low and low nodes are pulsed high. Holding the button down delivers a series of pulses of 20 pps to the circuit under test.

High input impedance (off state) 1 meg ohm • Multi mode-single pulses or pulse trains
Low output impedance (active state) 2 ohms • Automatic polarity sensing
Output pulse width 2 μsec nominal • Automatic current limiting; 7 amps nominal
Input over voltage protection +50 volts • Automatically programmed output level
Finger tip push button actuated • Circuit powered
Power lead reversal protection • No adjustments required
Multi-family RTL, DTL, TTL, CMOS, MOS and Microprocessors.

PRB 1	DIGITAL LOGIC PROBE	**$36.95**	PA 1	HIGH VOLTAGE ADAPTER	**$8.50**
PC 1	POWER CORD, Alligator Clips	**$4.95**	PT 2	REPLACEMENT PROBE TIP (2)	**$1.50**
PC 2	POWER CORD, Micro Hooks	**$9.95**	PLS 1	LOGIC PULSER	**$48.95**

OK Machine & Tool Corporation

54

3455 Conner St., Bronx, N.Y. 10475 U.S.A.

Tel. (212) 994-6600 Telex 125091

*Minimum billings $25.00, add shipping charge $2.00
New York State residents add applicable tax

```
*900. AFF
0900- 00 00 00 00 00 00 00 00
0908- 08 08 08 08 08 00 08 00
0910- 14 14 14 00 00 00 00 00
0918- 14 14 3E 14 3E 14 14 00
0920- 1C 2A 0A 1C 28 2A 1C 00
0928- 26 26 10 08 04 32 32 00
0930- 04 0A 0A 04 2A 12 2C 00
0938- 08 08 08 00 00 00 00 00
0940- 10 08 04 04 04 08 10 00
0948- 04 08 10 10 10 08 04 00
0950- 08 2A 1C 08 1C 2A 08 00
0958- 00 08 08 3E 08 08 00 00
0960- 00 00 00 00 10 10 08 00
0968- 00 00 00 3E 00 00 00 00
0970- 00 00 00 00 00 0C 0C 00
0978- 20 20 10 08 04 02 02 00
0980- 1C 22 32 2A 26 22 1C 00
0988- 08 0C 08 08 08 08 1C 00
0990- 1C 22 20 1C 02 02 3E 00
0998- 1E 20 20 1C 20 20 1E 00
09A0- 10 18 14 12 3E 10 10 00
09A8- 3E 02 1E 20 20 20 1E 00
09B0- 18 04 02 1E 22 22 1C 00
09B8- 3E 20 10 08 04 04 04 00
09C0- 1C 22 22 1C 22 22 1C 00
09C8- 1C 22 22 3C 20 10 0C 00
09D0- 00 0C 0C 00 0C 0C 00 00
09D8- 00 0C 0C 00 0C 0C 04 00
09E0- 10 08 04 02 04 08 10 00
09E8- 00 00 3E 00 3E 00 00 00
09F0- 04 08 10 20 10 08 04 00
09F8- 1C 22 10 08 08 00 08 00
0A00- 1C 22 3A 2A 3A 02 3C 00
0A08- 08 14 22 22 3E 22 22 00
0A10- 1E 24 24 1C 24 24 1E 00
0A18- 1C 22 02 02 02 22 1C 00
0A20- 1E 24 24 24 24 24 1E 00
0A28- 3E 02 02 1E 02 02 3E 00
0A30- 3E 02 02 1E 02 02 02 00
0A38- 3C 02 02 02 32 22 3C 00
0A40- 22 22 22 3E 22 22 22 00
0A48- 1C 08 08 08 08 08 1C 00
0A50- 20 20 20 20 22 22 1C 00
0A58- 22 12 0A 06 0A 12 22 00
0A60- 02 02 02 02 02 02 7E 00
0A68- 22 36 2A 2A 22 22 22 00
0A70- 22 26 2A 32 22 22 22 00
0A78- 1C 22 22 22 22 22 1C 00
0A80- 1E 22 22 1E 02 02 02 00
0A88- 1C 22 22 22 2A 12 2C 00
0A90- 1E 22 22 1E 0A 12 22 00
0A98- 1C 22 02 1C 20 22 1C 00
0AA0- 3E 08 08 08 08 08 08 00
0AA8- 22 22 22 22 22 22 1C 00
0AB0- 22 22 22 14 14 08 08 00
0AB8- 22 22 2A 2A 2A 36 22 00
0AC0- 22 22 14 08 14 22 22 00
0AC8- 22 22 22 1C 08 08 08 00
0AD0- 3E 20 10 08 04 02 3E 00
0AD8- 1C 04 04 04 04 04 1C 00
0AE0- 00 02 04 08 10 20 00 00
0AE8- 1C 10 10 10 10 10 1C 00
0AF0- 08 14 22 00 00 00 00 00
0AF8- 10 08 3E 7F FF FF BE 1C
```

Listing 3. Hex dump of the 512-byte standard character set. Enter using the Apple monitor. Actually, this set is not quite standard because code 223 is changed to an Apple character.

```
1000. 11FF
1000- 00 00 00 00 00 00 00 00
1008- 36 7F 7F 7F 3E 1C 08 00
1010- 08 1C 3E 7F 7F 1C 1C 00
1018- 08 1C 08 2A 7F 2A 08 00
1020- 08 1C 3E 7F 3E 1C 08 00
1028- 08 1C 2A 08 08 08 08 00
1030- 00 10 20 7F 20 10 00 00
1038- 08 08 08 08 2A 1C 08 00
1040- 00 04 02 7F 02 04 00 00
1048- 7C 40 48 44 7E 04 08 00
1050- 00 00 1C 36 14 14 3E 00
1058- 08 1C 08 1C 14 14 3E 00
1060- 08 2A 5D 49 49 2A 7F 00
1068- 18 24 2A 2E 28 28 3E 00
1070- 41 49 6B 55 55 55 7F 00
1078- 00 2A 3E 14 14 14 3E 00
1080- 00 08 00 3E 00 08 00 00
1088- 00 22 14 08 14 22 00 00
1090- 06 08 10 3C 22 22 1C 00
1098- 10 28 08 08 08 08 0A 04
10A0- 38 08 08 08 0A 0C 08 00
10A8- 00 00 3E 14 14 14 14 00
10B0- 00 1C 22 22 14 14 36 00
10B8- 08 1C 2A 2A 2A 1C 08 00
10C0- 00 1C 22 7F 22 1C 00 00
10C8- 49 2A 2A 2A 1C 08 08 00
10D0- 3E 22 04 08 04 22 3E 00
10D8- 00 20 1C 02 20 1C 02 00
10E0- 00 12 12 12 2E 02 02 00
10E8- 00 3C 0A 08 08 08 08 00
10F0- 00 4C 52 22 52 4C 00 00
10F8- 00 20 10 08 04 3E 00 00
1100- 0C 12 12 0C 00 00 00 00
1108- 00 00 1C 20 3C 22 3C 00
1110- 02 02 1A 26 22 22 1E 00
1118- 00 00 1C 22 02 22 1C 00
1120- 20 20 2C 32 22 22 3C 00
1128- 00 00 1C 22 3E 02 1C 00
1130- 18 24 04 0E 04 04 04 00
1138- 00 00 2C 32 22 3C 20 1E
1140- 02 02 1A 26 22 22 22 00
1148- 08 00 0C 08 08 08 1C 00
1150- 20 00 20 20 20 20 22 1C
1158- 02 02 12 0A 06 0A 12 00
1160- 00 0C 08 08 08 08 1C 00
1168- 00 00 16 2A 2A 2A 2A 00
1170- 00 00 1A 26 22 22 22 00
1178- 00 00 1C 22 22 22 1C 00
1180- 00 00 1E 22 22 1E 02 02
1188- 00 00 3C 22 22 3C 20 20
1190- 00 00 1A 26 02 02 02 00
1198- 00 00 3C 02 1C 20 1E 00
11A0- 04 04 1E 04 04 24 18 00
11A8- 00 00 22 22 22 32 2C 00
11B0- 00 00 22 22 14 14 08 00
11B8- 00 00 2A 2A 2A 2A 14 00
11C0- 00 00 22 14 08 14 22 00
11C8- 00 00 22 22 22 3C 20 1C
11D0- 00 00 3E 10 08 04 3E 00
11D8- 10 08 08 04 08 08 10 00
11E0- 08 08 08 00 08 08 08 00
11E8- 04 08 08 10 08 08 04 00
11F0- 00 04 2A 10 00 00 00 00
11F8- 2A 14 2A 14 2A 14 2A 00
```

Listing 4. Hex dump of the 512-byte lowercase special character set. The special characters include a mix of game, utility and scientific/math characters. Enter using the Apple monitor.

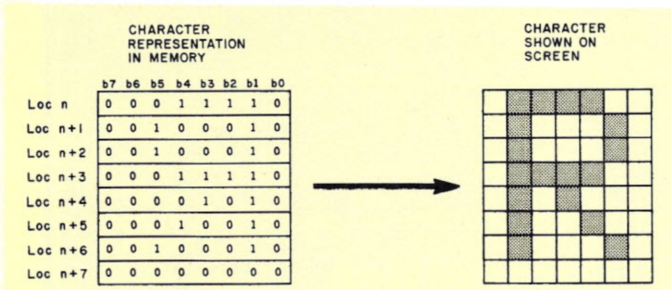


*Fig. 1. Within a character set, each character is represented by a block of eight bytes corresponding to the rows of dots in the character. Note that bit 0 represents the leftmost dot in a row, and bit 6 represents the rightmost. The location of the first byte in the set for a given character is: set start location + 8 * (character code — 160).*

Character Set Format

Characters on the Apple II are normally displayed as a 5 × 7 dot matrix within a 7 × 8 dot cell. There are 40 columns horizontally and 24 vertically, corresponding to a 280-dot horizontal by 192-dot vertical matrix. Coincidentally, this is the exact format of the hi-res screen, so it is easy for us to emulate normal text operation.

Within the character sets used by CHAR-GRAF, each character is represented by a group of eight bytes. Each byte corresponds to one row of dots in the character cell. For each character, the first byte represents the top row of dots in the cell, and the seven following bytes represent the second through eighth rows. Within each byte, bits 0-6 are mapped onto the screen as dots from left to right (see Fig. 1).

Character sets may contain from one to 96 characters. The first character in any set will be interpreted as a space; following characters will be interpreted in ascending ASCII order. (The first character in each set should be a blank because it is used by CHAR-GRAF to clear the rest of the line when a return code is received.)

Creating a character set by hand is awkward at best. However, the Character Set Editor program will allow you to create new character sets or edit existing ones with the Apple doing most of the work.

Getting CHAR-GRAF to Run

Type in the contents of Listing 3, check the contents of the last few locations and save the data on a tape using the Apple monitor. The same should be done for Listing 4 with a separate tape. These two tapes now hold the versions of the standard 64-character set and the 64-character lowercase, scientific and game character set.

Next, you should type in the hex contents of Listing 2, check your results with the Apple's disassembler and save the data on another section of tape. These are the actual CHAR-GRAF routines.

Now, load the Apple II hi-res graphics routines into memory at location $C00 as shown on the front of your hi-res demo cassette. The entire package of character sets and routines may now be saved to another section of tape with an *800.11FFW command. This creates an initial CHAR-GRAF package tape. The other tapes are backup, so that you can easily correct individual parts of the

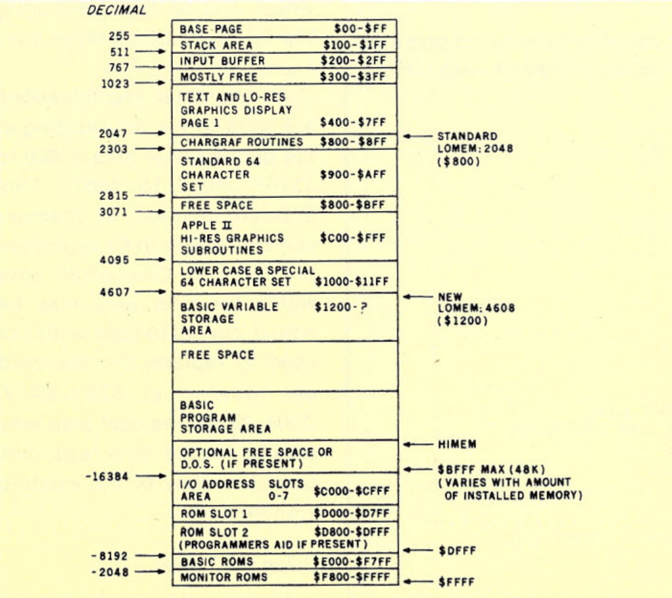


Fig. 2. Memory map of the Apple II with the CHAR-GRAF routines and two 64-character sets resident.

package.

Enter the contents of the demo program, load the CHAR-GRAF package tape and RUN the demo. If it runs properly, you're in good shape because it exercises all of the CHAR-GRAF functions. If not, at least you can have each section on a separate section of tape.

If there are mistakes in the CHAR-GRAF routines or the demo program, you will have to correct them now. Mistakes in the character sets will be easier to correct later after the Editor program is running.

Implementation

The STRIKEOVER function may be changed to an EXCLUSIVE-OR function by changing location $86D from $01 to $51. This can be done from BASIC with a POKE 2157,81, and changed back with a POKE 2157,1. (Be careful! You are actually modifying the CHAR-GRAF code.)

To use CHAR-GRAF with hi-res routines in Apple's Programmer's Aid #1 ROM, you must make the changes in the box below to CHAR-GRAF.

The June '79 issue of *CALL A.P.P.L.E.* contains a machine-language linker program by Andy Hertzfeld that appears to be the best available way of attaching machine-language routines to an Apple Integer BASIC program. I use this routine to save the CHAR-GRAF package along with the BASIC program using it.

Note that CHAR-GRAF uses the same screen pointer locations used by the Apple hi-res routines. This means that after using CHAR-GRAF to print some text, the hi-res position will be left somewhere in the bottom row of the last character printed, and the old position will be forgotten.

Hex Loc.	From	To
$85D	$2E	$26
$891	$2E	$26
$8B2	$2C	$24
$8B7	$2C	$24

The Demo Program

The demo program, written in BASIC, demonstrates all of the current CHAR-GRAF functions. It requires the Apple hi-res graphics routines and both character sets to be resident in memory with it. Notice that most of the PEEK, POKE and CALL locations are assigned as values to descriptive variable names. This allows the use of such statements as:

CALL STRIKEOVER and POKE XL0,5

rather than

CALL 2180 and POKE 800,5

This nice feature of Apple's BASIC allows programs to be more self-documenting than usual and easier to write. Useful routines in the demo program are found at lines 360, 450 and 570. Comments are included.

Character Set Editor

The Character Set Editor program was a result born of my frustration with creating character sets by hand. I had invariably made mistakes—either typos, dropped bytes on input or, in one case, a set completely transposed left to right. (I wrote a short program to correct that one!)

I decided to write the editor to do as much work as possible for me. It took much longer to write than it took to write CHAR-GRAF in the first place, but the time spent was worth it. I could now create a whole new character set in about a half hour with no errors.

Editor Description

The editor is designed to allow the user to manipulate character sets in variety of ways. Its capabilities include:

- Storage and retrieval of character sets from tape or disk
- Editing existing character sets
- Creating new character sets
- Displaying character sets
- Creating CHAR-GRAF packages on tape with one or two character sets included

The editor is currently designed to handle sets of up to 64 characters. However, larger sets are easy to create in sections,

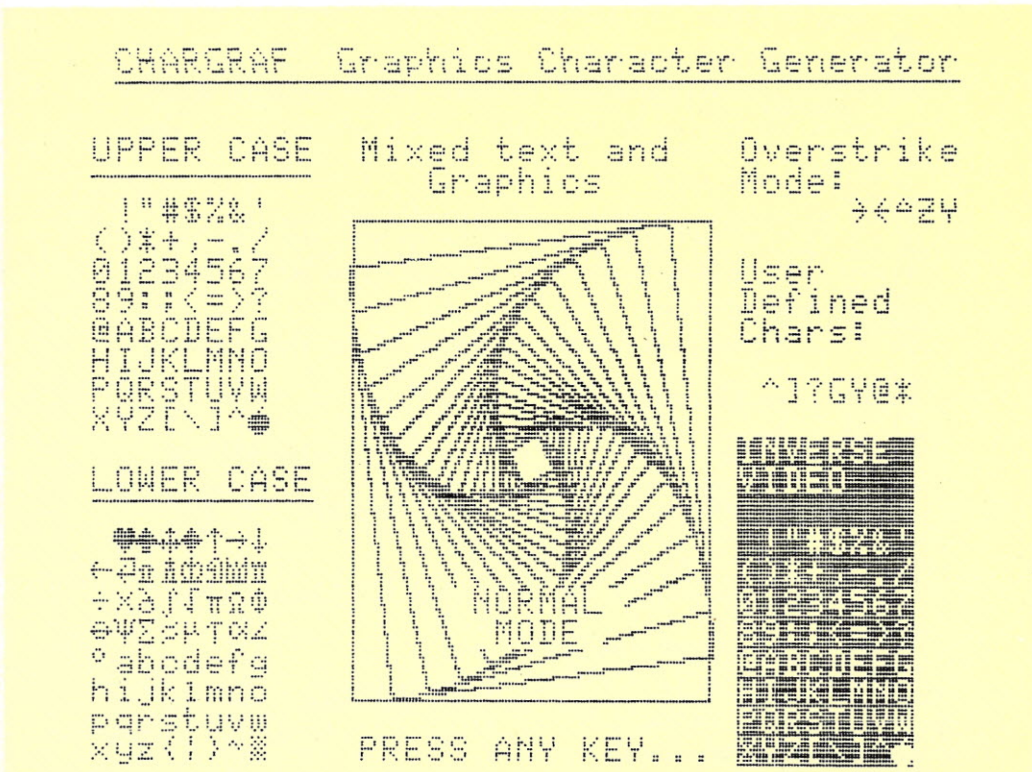


Fig. 3. Display produced by the demo program showing the various CHAR-GRAF features.

Listing 5. CHAR-GRAF demo program, which creates screen display using both character sets and exercises all of the CHAR-GRAF functions. It also serves as an example of ways to use CHAR-GRAF.

```
  0 REM ****************************
 10 REM *                          *
 20 REM *  COPYRIGHT 1979          *
 30 REM *  R. B. MOORE             *
 40 REM *                          *
 50 REM *  ALL COMMERCIAL RIGHTS   *
 60 REM *  RESERVED                *
 70 REM *                          *
 80 REM ****************************
 85 GOSUB 580: REM **SET LOMEM: 4608
 90 GOSUB 500: REM ** INIT PROG
 95 REM
100 REM  ** DO DEMO **
105 REM
110 CALL INIT: POKE -16302,0: CALL NONINV: CALL HOME: CALL LINK
120 TAB 2: PRINT "CHARGRAF  ";: T$="GRAPHICS ": GOSUB 450: T$="CHARACTER "
130 GOSUB 450: T$="GENERATOR": GOSUB 450
135 REM
140 REM  ** DRAW LINE UNDER TITLE **
145 REM
150 POKE HY, 9: POKE XHI, 0: POKE XLO, 7: CALL POSN: POKE XHI, 1: POKE XLO,
    17: CALL LINE
155 REM
160 REM  ** DO SUBTITLE **
165 REM
170 VTAB 4: TAB 13: T$="MIXED": GOSUB 450: T$=" TEXT AND": GOSUB 450
180 VTAB 5: TAB 16: T$="GRAPHICS": GOSUB 450
185 REM
190 REM  ** DO U. C. & L. C. SETS **
195 REM
200 VTAB 6: TAB 1: GOSUB 470: VTAB 17: TAB 1: POKE TH, 16: GOSUB 470
205 REM
210 REM ** DO U. C. AND L. C. TITLES
215 REM
220 POKE TH, 9: VTAB 4: TAB 1: PRINT "UPPER CASE";: VTAB 15: TAB 1: PRINT
    "LOWER CASE";
225 REM
230 REM  ** UNDERLINE TITLES **
235 REM
240 POKE XHI, 0: POKE XLO, 0: POKE HY, 34: CALL POSN: POKE XLO, 69: CALL LINE
250 POKE HY, 121: CALL POSN: POKE XLO, 0: CALL LINE
255 REM
260 REM  ** DO INVERSE VIDEO STUFF **
265 REM
270 CALL INV: VTAB 14: TAB 30: PRINT "INVERSE ";: VTAB 15: TAB 30
280 PRINT "VIDEO   ";: VTAB 16: TAB 30: PRINT "        ";
290 VTAB 17: TAB 30: GOSUB 470: CALL NONINV
295 REM
300 REM  ** SHOW USER CHARACTERS **
305 REM
310 VTAB 8: TAB 30: T$="USER": GOSUB 450: VTAB 9: TAB 30: T$="DEFINED": GOSUB
    450
320 VTAB 10: TAB 30: T$="CHARS": GOSUB 450: PRINT ":";: VTAB 12: TAB 31
    : PRINT "^]?GY@*"
325 REM
330 REM  ** OVERSTRIKE DEMO **
335 REM
340 VTAB 4: TAB 30: T$="OVERSTRIKE": GOSUB 450: VTAB 5: TAB 30: T$="MODE"
    : GOSUB 450: PRINT ":";
350 VTAB 6: TAB 35: PRINT "><^ZY";: VTAB 6: TAB 35: CALL STRIKEOVER: PRINT
    "-----";: CALL LINK
355 REM
360 REM  ** DO PICTURE **
365 REM
370 FOR I=0 TO 3: X(I)=56-112*(I/2): Y(I)=64-128*((I MOD 3)>0): NEXT I
380 FOR J=1 TO 20: POKE HY, 110+Y(3): POKE XLO, 138+X(3): CALL POSN
390 FOR I=0 TO 3: POKE HY, 110+Y(I): POKE XLO, 138+X(I): CALL LINE
400 K=(X(I)+Y(I)/8)*9/10: Y(I)=(Y(I)-X(I)/8)*9/10: X(I)=K: NEXT I, J
405 REM
410 REM  ** FINISH UP **
415 REM
420 VTAB 19: TAB 17: PRINT " NORMAL ";: VTAB 20: TAB 18: PRINT " MODE "
    ;
425 REM
430 REM  ** WAIT FOR KEYSTROKE **
435 REM
440 VTAB 24: TAB 13: PRINT "PRESS ANY KEY...";: K=0
445 IF PEEK (-16384)<128 THEN 445: POKE -16368, 0: IF K=1 THEN 449: FOR
    I=1 TO 11
448 CALL INVSCRN: PRINT "";: FOR J=1 TO 100: NEXT J, I: K=1: GOTO 445
449 CALL INVSCRN: FOR J=1 TO 500: NEXT J: TEXT : CALL HOME: PR#0: END
450 REM
451 REM   ** PRINT T$ W/ 1ST CHAR CAP **
455 REM
460 POKE TH, 9: PRINT T$(1,1);: POKE TH, 16: PRINT T$(2, LEN(T$));: POKE
    TH, 9: RETURN
465 REM
470 REM  ** PRINT CURRENT CHARACTER SET **
475 REM
480 FOR I=0 TO 7: FOR J=0 TO 7: POKE CHARLOC, 160+I*8+J
490 CALL CHARDRAW: NEXT J: IF I<>7 THEN VTAB ( PEEK (37)+2): TAB PEEK
    (36)-7: NEXT I: RETURN
495 REM
500 REM  ** SETUP AND DIMS **
505 REM
510 TL=816: TH=817: PG1=32: PG2=64: CHARLOC=71: CHARDRAW=2172: STRIKEOVER=2180
    : INVSCRN=2187
```

which may be appended using the monitor's Block Move command.

All character manipulations are done to the set residing in the defined user area \$1000 to \$11FF (4096 to 4607). This 512-byte section of memory can contain up to 64 eight-byte characters. Character sets within the user area may be edited, copied to tape or disk or used to replace the standard set located at \$900–\$AFF (2304–2815). The user area may also be loaded from tape, disk or with a copy of the existing standard set.

Using the Editor

After the editor is typed in and saved on tape, reload the CHAR-GRAF packages tape and RUN the editor. A menu of the various editor options should appear: An explanation of each option follows.

1. Edit/Create character set. The EDIT/CREATE mode allows you to create or modify character sets by moving a cursor around an enlarged 7 × 8 dot grid. When the option is selected, the Apple is put into hi-res mode and a titled map of the character set in the user area is displayed along with the edit box and a list of edit commands.

To start, you must enter the code of the first character that you wish to edit. A small box will appear around that character in the map, and an enlarged version of the character will appear in the edit box. The available edit commands are as follows:

<- -	Move cursor left
- ->	Move cursor right
RETURN	Move cursor down
/	Move cursor up
D	Change state of current dot
N	Done, go to next character
P	Done, go to previous character
C	Clear grid in edit box
A	Done, accept new code
R	Restart edit of this character
M	Done, go back to menu

When you are done with a particular character, the contents

```
520 INIT=3072: LINK=2048: HOME=-936: LINE=3786: POSN=3761. PLOT=3780: HY=802
    : XLO=800: XHI=801: HCOL=812
530 INV=-384: NONINV=-380
540 POKE TL, 0:  POKE TH, 9:  POKE HCOL, 255
550 DIM T$(30), X(4), Y(4)
560 RETURN
565 REM
570 REM  ** SET LOMEM: 4608 **
575 REM
580 POKE 74, 4608 MOD 256:  POKE 75, 4608/256:  POKE 204, 4608 MOD 256:  POKE
    205, 4608/256:  RETURN
```

of the edit box are transferred back into the character set if the character has been altered. If you change your mind about an edit, press R before exiting the character. This restores the original character and nullifies any changes that you may have made. When you are through, press M to return to the editor menu.

2. Load character set. Character sets may be loaded from tape or disk. When loading from disk, the editor will automatically do a disk CATALOG and then ask for the filename. Make sure that the file you select is a character set; otherwise, the program may be destroyed. After the filename is supplied, the program will ask if you wish to load into the user area or replace the standard set. Reply by pressing S or U. Loading from tape loads into the user area only.

3. Save character set. The set in the user area may be saved to tape or disk. If the disk option is selected, the program will ask for a filename before saving to disk.

4. Create CHAR-GRAF package. This option creates a package tape that may be used with one of your programs. The package includes CHAR-GRAF, the standard set, the Apple hi-res routines and, optionally, the user set.

5. Display hi-res character sets. Draws the character set map and alternately displays the standard set and the user set when a key is pressed. Pressing RETURN will set you back to the menu.

6. User set → std set.

7. Std set → user set. These two options copy one set to the other. Note that this is a copy, *not* a swap! After using either option, you will have the same character set in both places.

8. Exit program. This unlinks CHAR-GRAF and ends the program. The package of routines and character sets is not affected, and LOMEM is left set at 4608. The program may be restarted again with the same sets.

General Comments

In *Apple Computer's User Contributed Software,* vols. 3 & 5, there is a character generator program that uses a 128-character set and an Applesoft II demo program to go with it. Coincidentally, the character format is exactly the one I used. This means that you can use the Character Set Editor with that program also.

Some manipulation of partial sets will be required to create the 128-character sets, but it's not difficult. Also, that set is easy to divide into two parts to use with CHAR-GRAF.

The Character Set Editor occupies approximately 7.5K bytes and requires at least 36K of memory if used with a disk system or 24K in a tape-based system. Removing all of the REM statements from the program will reduce it to about 5.6K. It could be reduced further by combining short lines and shortening the printed messages.

Possible Uses for CHAR-GRAF

In addition to the obvious uses titling plots and graphs, there are many other possibilities. A friend of mine has used CHAR-GRAF to create an I.Q. Test program that uses shape and figure analogies drawn on the hi-res screen along with the text. You could use CHAR-GRAF to create chessmen out of four or nine characters each and PRINT them onto a chessboard using the exclusive-OR mode described earlier. White pieces could be PRINTed normally, and black pieces could be in inverse-video mode.

A set of playing card characters could be created to add realism to card games such as poker or blackjack (see Bill Depew's Apple 21, from Softape, Inc., using the Screen Machine). For hi-res graphics games you could design ships, planes, tanks or tie-fighters in various rotations. These could be placed on the screen with PRINT statements (and erased with another print if the exclusive-OR mode is used). Programs for the PET computer could be directly converted to the Apple by creating a "PET SET."

Other Alternatives

In recent months, several alternative methods of hi-res character generation have emerged from various Apple after-market suppliers: the Screen Machine ($19.95), Softape, Inc.; the SuperChip (ROM) ($94.95) and Editor ($19.95), Eclectic Enterprises, Inc.; and the ROMPLUS + (board) ($169), Mountain Hardware, Inc.

For those of you who just want lowercase, there are several lowercase adapters that retail in the $50 range in local computer stores. ■

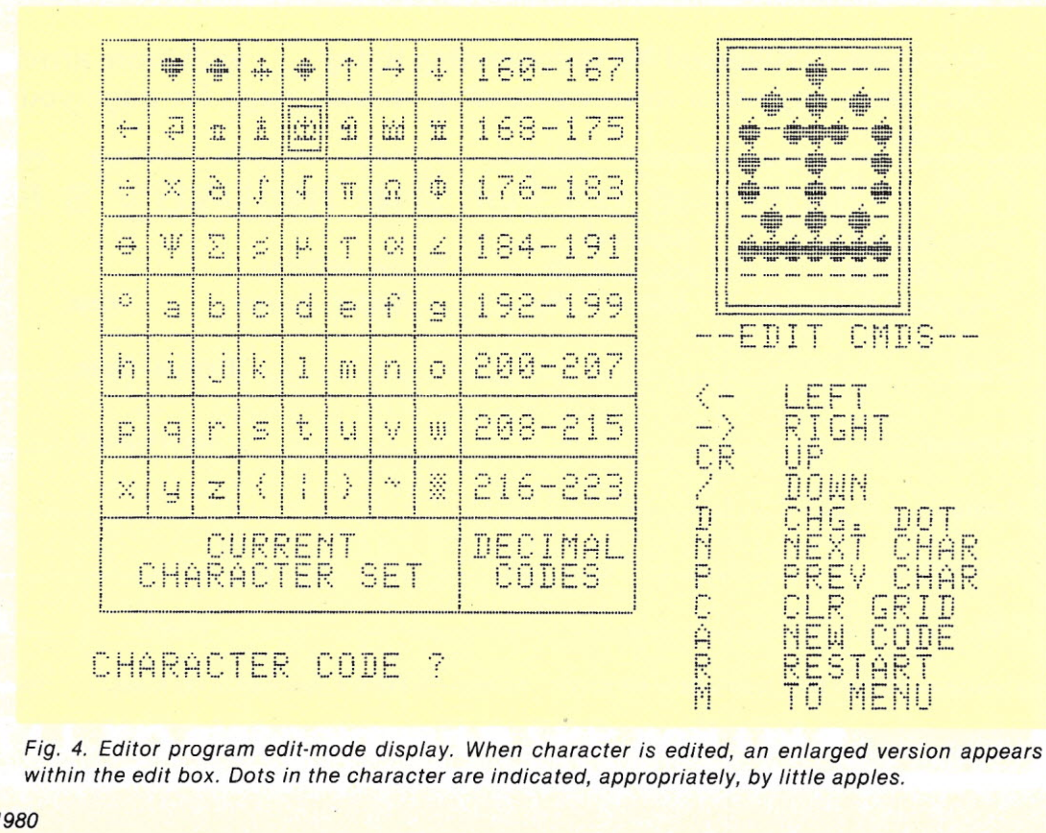


Fig. 4. Editor program edit-mode display. When character is edited, an enlarged version appears within the edit box. Dots in the character are indicated, appropriately, by little apples.

Listing 6. Character Set Editor program, which allows you to create and edit your own character sets with the Apple doing most of the work for you. It requires a 24K system (or 36K, if you have a disk).

```
 10 REM ***************************
 20 REM *                         *
 30 REM *  COPYRIGHT 1979         *
 40 REM *  R. B. MOORE            *
 50 REM *                         *
 60 REM *  ALL COMMERCIAL RIGHTS  *
 70 REM *  RESERVED               *
 80 REM *                         *
 90 REM ***************************
100 REM INIT LOMEM&VARIABLES&DO MENU
110 GOSUB 1920: GOSUB 1660: GOTO 1790
120 REM
130 REM XFER EDIT GRID TO CHAR SET
140 REM
150 IF FLG=0 THEN RETURN
160 FOR I=0 TO 7:K=0: FOR J=6 TO 0 STEP -1:VL=I+2:HL=J+30: GOSUB 220
170 K=K+K+(CHAR=223): NEXT J: POKE 4096+I+CCHAR*8,K: NEXT I
180 POKE TH,16: GOSUB 750: POKE TH,9: RETURN
190 REM
200 REM  PICKUP CHAR FROM SCREEN
210 REM
220 VTAB VL:CHAR= PEEK ( PEEK (40)+ PEEK (41)*256+HL-1): RETURN
230 REM
240 REM  CURRENT CHAR TO EDIT GRID
250 REM
260 CALL LINK: GOSUB 620: POKE CHARLOC,223: FOR I=X TO 7
270 J= PEEK (4096+I+CCHAR*8):L=64
280 IF J>127 THEN J=J-128
290 FOR K=6 TO 0 STEP -1: IF J<L THEN 310
300 J=J-L: VTAB (I+2): TAB (30+K): CALL CHARDRAW
310 L=L/2: NEXT K,I: RETURN
320 REM
330 REM PSEUDO-CURSOR AND KEY INPUT
340 REM CALL WITH VL(0-23),HL(0-39)
350 REM RETURNS KEYSTROKE IN 'KEY'
360 REM
370 FLASH=0: GOSUB 220: POKE CHARLOC,CHAR:I=0
380 I=(I+1) MOD 6: IF I=0 THEN FLASH=1-FLASH: POKE 50,63+192*FLASH: VTAB
     VL: TAB HL: CALL CHARDRAW
390 IF PEEK (-16384)<128 THEN 380: POKE CHARLOC,CHAR: POKE 50,255: VTAB
     VL: TAB HL: CALL CHARDRAW: GOSUB 810: RETURN
400 REM
410 REM *** SETUP CHAR SET MAP ***
420 REM
430 CALL INIT: POKE HCOL,255: FOR I=0 TO 8: POKE XLO,3: POKE XHI,0: POKE
     HY,(3+16*I): CALL POSN
440 POKE XLO,171: CALL LINE: NEXT I: POKE HY,157: POKE XLO,3: CALL POSN:
     POKE XLO,171: CALL LINE
450 FOR I=0 TO 8: POKE XLO,(3+14*I): POKE HY,3: CALL POSN: POKE HY,131
     : CALL LINE: NEXT I
460 POKE XLO,171: POKE HY,3: CALL POSN: POKE HY,157: CALL LINE
470 POKE XLO,115: CALL POSN: POKE HY,131: CALL LINE: POKE XLO,3: CALL
     POSN: POKE HY,157: CALL LINE
480 REM
490 REM LINK TO CHARGRAF AND FILL IN
500 REM  CHART TITLES...
510 REM
520 CALL LINK: VTAB 18: TAB 6: PRINT "CURRENT";: TAB 18: PRINT "DECIMAL"
     ;: VTAB 19: TAB 3: PRINT "CHARACTER SET";: TAB 19: PRINT "CODES";
530 FOR I=0 TO 7: VTAB 2*(I+1): TAB 18: PRINT I*8+160;"-";I*8+167;: NEXT
     I: RETURN
540 REM
550 REM  *** DRAW EDIT BOX ***
560 REM
570 POKE HY,3: POKE XLO,196: POKE XH1,0: CALL POSN: POKE XLO,1: POKE XHI,
     1: CALL LINE: POKE HY,77: CALL LINE
580 POKE XHI,0: POKE XLO,196: CALL LINE: POKE HY,3: CALL LINE
590 POKE HY,5: POKE XLO,199: CALL POSN: POKE XLO,254: CALL LINE: POKE
     HY,75: CALL LINE
600 POKE XLO,199: CALL LINE: POKE HY,5: CALL LINE: RETURN
610 REM
620 REM  *** FILL IN EDIT BOX ***
630 REM
640 FOR I=2 TO 9: VTAB (I): TAB (30): PRINT "-------";: NEXT I: VTAB 24
     : RETURN
650 REM
660 REM SHOW OR DROP EDIT CHAR
670 REM
680 POKE HCOL,255*FLG: POKE XHI,0:I=(CCHAR MOD 8+1)*14-9
690 J=(CCHAR/8+1)*16-11: POKE XLO,I: POKE HY,J: CALL PLOT
700 POKE XLO,I+10: CALL LINE: POKE HY,J+12: CALL LINE: POKE XLO,I: CALL
     LINE: POKE HY,J: CALL LINE
710 RETURN
720 REM
730 REM *** FILL IN CHARACTER CHART
740 REM
750 FOR I=0 TO 63: TAB (I MOD (8)*2+2): VTAB (I/8*2+2)
760 POKE CHARLOC,160+I: CALL CHARDRAW
770 NEXT I: VTAB 21: TAB 1: RETURN
780 REM
790 REM  ** PICKUP KEY STROKE **
800 REM
810 KEY= PEEK (-16384): IF KEY<128 THEN 810: POKE -16368,0: RETURN
820 REM
830 REM **DISPLAY HIRES CHAR SETS
840 REM
```

```
850 POKE TH,9: GOSUB 430
860 POKE TH,9: GOSUB 750: CALL HOME: VTAB 21: PRINT "STANDARD CHARACTER SET
    "
870 PRINT : PRINT "HIT 'RETURN TO EXIT TO MENU, ANY OTHER  KEY TO ALTERNATE
     SETS";:S=1
880 GOSUB 810: IF KEY=141 THEN 1790: IF S=0 THEN 860
890 POKE TH,16: GOSUB 750: VTAB 21: PRINT "USER DEFINED CHARACTER SET"
    :S=0: GOTO 880
900 REM
910 REM ** SAVE CHARGRAF PKG. **
920 REM
930 CALL HOME: PRINT "  *** SAVE CHARGRAF PACKAGE ON TAPE ***": VTAB 3

940 PRINT "OPTION 1:": PRINT "---------": PRINT . PRINT "SAVES";: TAB
    17: PRINT "LOC. (HEX) * LOAD USING"
950 PRINT "-----";: TAB 17: PRINT "--------- * MONITOR": TAB 27: PRINT
    "*": PRINT "CHARGRAF";: TAB 17: PRINT "800-8DF   * 800.FFFR"
960 PRINT "STD. CHAR. SET  900-AFF   *": PRINT "UNUSED AREA     BOO-BFF   *
     SET BASIC"
970 PRINT "HIRES ROUTINES  COO-FFF   * LOMEM:4096": PRINT : PRINT : PRINT
    "OPTION 2:": PRINT "---------": TAB 27: PRINT "* LOAD USING"
980 PRINT "SAVES ALL OF OPTION 1      * 800.11FFR": PRINT "AND INCLUDES THE
    USER-    *": PRINT "DEFINED CHARACTER SET.    * SET BASIC"
990 PRINT "     (1000-11FF HEX)";: TAB 27: PRINT "* LOMEM:4608": PRINT
    : PRINT "ENTER OPTION #, START RECORDING, AND    PRESS 'RETURN'";
1000 INPUT I: IF I<1 OR I>2 THEN 1790: CALL HOME: VTAB 13: PRINT "SAVING DAT
     A": IF I=2 THEN 1020
1010 K$="800.FFFW N E88AG": GOTO 1600
1020 K$="800.11FFW N E88AG": GOTO 1600
1030 REM
1040 REM *** SAVE USER SET ***
1050 REM
1060 CALL HOME: VTAB 10: PRINT "SAVE USER SET TO TAPE OR DISK (T/D)?": GOSUB
     810: IF KEY=212 THEN 1090
1070 IF KEY#196 THEN 1790: CALL HOME: VTAB 10: INPUT "ENTER FILENAME FOR THI
     S SET: ",N$: IF N$="" THEN 1790
1080 PRINT : PRINT D$;"BSAVE";N$;",A$1000,L$200,V0": GOTO 1790
1090 CALL HOME: PRINT "SAVE USER DEFINED CHARACTER SET ON TAPE": VTAB 5

1100 PRINT "THIS TAPE MAY BE LOADED WITH THIS PROG- RAM OR WITH THE APPLE MO
     NITOR"
1110 PRINT : PRINT "TO LOAD WITH THE MONITOR:": PRINT : PRINT "TYPE '1000.11
     FFR', START THE TAPE,       AND PRESS 'RETURN'"
1120 VTAB 17: PRINT "TO SAVE THE DATA, START RECORDING        AND PRESS 'RETU
     RN'"
1130 GOSUB 810: IF KEY-128<>13 THEN 1790
1140 CALL HOME: VTAB 12: PRINT "RECORDING DATA ON TAPE":K$="1000.11FFW N E88
     AG": GOTO 1600
1150 REM
1160 REM ** LOAD CHARACTER SET **
1170 REM
1180 CALL HOME: VTAB 10: PRINT "LOAD SET FROM TAPE OR DISK (T/D)?": GOSUB
     810
1190 IF KEY=212 THEN 1230: IF KEY#196 THEN 1790: CALL HOME: PRINT D$;"CATALO
     G": PRINT : INPUT "SET FILENAME ?",N$
1200 IF N$="" THEN 1790: PRINT : PRINT "STD SET OR USER SET (S/U)? ": GOSUB
     810: IF KEY=213 THEN 1220
1210 IF KEY#211 THEN 1790: PRINT D$;"BLOAD";N$;",A$900": GOTO 1790
1220 PRINT D$;"BLOAD";N$;",A$1000": GOTO 1790
1230 CALL HOME: PRINT "READ CHARACTER SET TAPE": VTAB 6
1240 PRINT "THIS ROUTINE READS A 512 BYTE CHARACTER SET TAPE INTO THE USER-D
     EFINED SET AREA."
1250 PRINT "START THE TAPE PLAYING": PRINT "AND PRESS 'RETURN'": GOSUB
     810: IF KEY<>13 THEN 1790
1260 K$="1000.11FFR N E88AG": CALL HOME: VTAB 13: PRINT "READING IN DATA"
     : GOTO 1600
1270 REM
1280 REM EDIT/CREATE NEW USER SET
1290 REM
1300 CALL HOME
1310 POKE TH,9: GOSUB 430: POKE TH,16: GOSUB 750: GOSUB 570: POKE TH,9
1320 POKE 33,13: POKE 32,27
1330 VTAB 11: PRINT "--EDIT CMDS--"
1340 VTAB 13: PRINT "<-  LEFT": PRINT "->  RIGHT": PRINT "CR  UP": PRINT
     "/   DOWN"
1350 PRINT "D   CHG. DOT": PRINT "N   NEXT CHAR";: PRINT "P   PREV CHAR"
     ;: PRINT "C   CLR GRID": PRINT "A   NEW CODE"
1360 PRINT "R   RESTART": PRINT "M   TO MENU"
1370 POKE 33,27: POKE 32,0: POKE 35,23: VTAB 22: TAB 1: INPUT "CHARACTER COD
     E ",CCHAR
1380 POKE 34,0: CALL HOME:CCHAR=CCHAR-160
1390 IF CCHAR<0 OR CCHAR>63 THEN 1790: REM  BAD CODE, BACK TO MENU
1400 VTAB 21: TAB 1: PRINT "ASCII CODE = ";CCHAR+160;: POKE 33,40: POKE
     35,20: CALL HOME
1410 FLG=1: GOSUB 680: GOSUB 260:D=0:FLG=0: GOTO 1430
1420 IF KEY>192 THEN 1440:D=D+(KEY=149)+55*(KEY=136)+7*(KEY=141)+49*(KEY=
     175): REM  ALTER GRID#
1430 D=D MOD 56:HL=30+D MOD 7:VL=2+D/7: GOSUB 370: GOTO 1420: REM  POSN CURS
     OR AND GET KEY
1440 IF KEY#208 THEN 1450: GOSUB 150:FLG=0: GOSUB 680:CCHAR=CCHAR-(CCHAR>
     0): GOTO 1400: REM  PREV CHAR
1450 IF KEY#206 THEN 1460: GOSUB 150:FLG=0: GOSUB 680:CCHAR=CCHAR+(CCHAR<
     63): GOTO 1400: REM  NEXT CHAR
1460 IF KEY#196 THEN 1490
1470 GOSUB 220:CHAR=173+50*(CHAR=173): POKE CHARLOC,CHAR: VTAB VL: TAB
     HL:FLG=1: REM  ALTER DOT
1480 CALL CHARDRAW:D=D+1: GOTO 1430
1490 IF KEY#195 THEN 1500: GOSUB 640:FLG=1:D=0: GOTO 1430: REM  CLR GRID
1500 IF KEY#210 THEN 1510: GOTO 1410: REM  RESTART CHAR
1510 IF KEY#193 THEN 1520: GOSUB 150:FLG=0: GOSUB 680: GOTO 1370: REM  ACCEP
     T NEW CODE
1520 IF KEY#205 THEN 1430
1530 GOSUB 150: GOTO 1790: REM  M FOR MENU
1540 REM. STD SET <-- USER SET
1550 K$="900<1000.11FFM N E88AG": GOTO 1600
1560 REM  USER SET <-- STD SET
```

```
1570 K$="1000<900. AFFM N E88AG"
1580 REM
1590 REM ** EXEC A MONITOR CMD. **
1600 FOR I=1 TO LEN(K$): POKE 511+I, ASC(K$(I,I)): NEXT I: POKE 72,0: CALL
     -144
1610 GOTO 1790
1620 END
1630 REM
1640 REM  DEFINE USEFULL CONSTANTS
1650 REM
1660 I=J=K: PAGE=814: TBLE=2304: DIM K$(40),N$(26): D$="": REM  CNTRL-D
1670 TL=816: TH=817: PG1=32: PG2=64: CHARLOC=71: CHARDRAW=2172
1680 INIT=3072: LINK=2048: HOME=-936: LINE=3786: POSN=3761: PLOT=3780: HY=802
     : XLO=800: XHI=801: HCOL=812
1690 INV=-384: NONINV=-380
1700 REM
1710 REM SETUP CHAR TABLE PPOINTERS
1720 REM   AND HIRES PAGE
1730 REM
1740 POKE TL,0: POKE TH,9: POKE PAGE,PG1
1750 RETURN
1760 REM
1770 REM  HANDLE MENU AND CHOICE
1780 REM
1790 TEXT : PR#0: CALL -936: TAB 5: PRINT " *** CHARACTER SET EDITOR ***"
     : VTAB 4
1800 PRINT "1.   EDIT/CREATE CHARACTER SET": PRINT : PRINT "2.   LOAD CHARACTE
     R SET": PRINT
1810 PRINT "3.   SAVE CHARACTER SET": PRINT : PRINT "4.   CREATE CHAR-GRAF PAC
     KAGE": PRINT
1820 PRINT "5.   DISPLAY HIRES CHARACTER SETS": PRINT
1830 PRINT "6.   USER SET --> STD. SET. ": PRINT : PRINT "7.   STD. SET --> USE
     R SET. ": PRINT
1840 PRINT "8.   EXIT FROM PROGRAM": PRINT
1850 PRINT : PRINT "ENTER THE NUMBER DESIRED": INPUT "AND PRESS 'RETURN' "
     ,F: IF F<1 OR F>8 THEN 1790
1860 IF F=1 THEN 1300: IF F=2 THEN 1180: IF F=3 THEN 1060: IF F=4 THEN
     930
1870 IF F=5 THEN 850: IF F=6 THEN 1550: IF F=7 THEN 1570: TEXT : PR#0: CALL
     HOME
1880 PRINT "IF YOU WANT TO SAVE YOUR CHARACTER SET  TYPE 'RUN' AND PRESS 'RE
     TURN'": END
1890 REM
1900 REM ** SET LOMEM AND V. P.
1910 REM
1920 POKE 74,4608 MOD 256: POKE 75,4608/256: POKE 204,4608 MOD 256: POKE
     205,4608/256: RETURN
```

Thoughts on the SWTP Computer System

Don't be a scrooge; catch the true spirit of this series with the HUMBUG monitor.

Peter A. Stark
PO Box 209
Mt. Kisco, NY 10549

This month, we continue the ROM monitor discussion we started last month. The first ROM monitor in SWTP systems was MIKBUG. Most software was designed to work with it, and so succeeding monitors have had to copy many of MIKBUG's routines and addresses.

The important MIKBUG entry points, which should be preserved in "compatible" monitors are:

BADDR	E047—Input four hex digits into index register
BYTE	E055—Input two hex digits into A accumulator
OUTHL	E067—Output left BCD digit in A accumulator
OUTHR	E06B—Output right BCD digit in A accumulator
OUTCH	E075—Points to OUTEEE
INCH	E078—Points to INEEE
PDATA1	E07E—Print a text string pointed to by index reg.
INHEX	E0AA—Input a hex digit into A accumulator
OUT2H	E0BF—Output two hex digits pointed to by index reg.
OUT4HS	E0C8—Output four hex digits pointed to by index reg., followed by a space
OUT2HS	E0CA—Output two hex digits pointed to by index reg., followed by a space
OUTS	E0CC—Print a space
START	E0D0—Start MIKBUG
CONTRL	E0E3—Restart MIKBUG
INEEE	E1AC—Input a 7-bit character from keyboard
OUTEEE	E1D1—Output a character to terminal

These sixteen entry points are the major ones. In addition, there are about 20 more minor ones that you can include if you just copy most of MIKBUG, but which are probably otherwise not needed.

The one exception is the SWTP BILOAD program, which is used to speed up loading of binary tapes such as BASIC. This program uses these additional MIKBUG entry points:

DMPREG	E115—Print out CPU registers
LOAD19	E040—Part of load routine
SAV	E1A5
DE	E1F3
DEL	E1EF
IOUT2	E1E3

This loader does not work with an MP-S interface, so I chose not to include these entry points. However, I did include an entry point called INCH8 at E1F6, which is similar to INEEE except that it enters an 8-bit ASCII character rather than stripping off the parity bit to make it into seven bits, as INEEE does.

MIKBUG also uses the 128-byte scratchpad RAM starting at location A000. There are some differences, however, between MIKBUG and SWTBUG in address assignments in this area, and I chose to go with SWTBUG here rather than with MIKBUG. The important addresses are as follows:

IRQ	A000—IRQ interrupt vector
BEGA	A002—Beginning address for punching, etc.
ENDA	A004—Ending address for punching, etc.
NMI	A006—NMI interrupt vector
SP	A008—User stack pointer
PORADD	A00A—Address of the control port in use
PORECH	A00C—Terminal echo on/off flag
XHI	A00D—High-order half of index register
XLOW	A00E—Low-order half of index register
CKSM	A00F—Checksum
SWIJMP	A012—SWI interrupt vector
PC	A048—Program counter for Go command

MIKBUG had XHI and XLOW one location lower, and some other monitors (as well as some user software) go along with this convention.

I also treated the stack differently. MIKBUG and SWTBUG always initialize the stack when they are started up at A042 and down. The G command then loads the next seven bytes into CPU registers and jumps to a user program with the stack pointer pointing to A049. So, in a way, we can think of the area below A042 as being a monitor stack, while the area just below A049 is a user stack.

But SWTBUG's J command doesn't change the stack pointer when going to a user program; it leaves it pointing to the monitor area. Likewise, when a breakpoint is encountered, it leaves the stack pointer unchanged when it executes its own routines. This results in some weird occurrences when the monitor and user stacks wipe each other out. It becomes even more interesting when you consider that some user software initializes the stack elsewhere . . . such as at A042.

Because of this, I put other HUMBUG storage locations in a separate RAM—far away from the MIKBUG/SWTBUG RAM—and treated the stacks differently. The monitor stack is now always at D07F. A jump command always goes to a user program with the stack pointer at A07D (with a return address at A07E/F, so jumping to a subroutine will result in a return back to the monitor), and a GO command always goes to the user program with the stack pointer at A049.

This keeps monitor and user stacks completely separate so they never clobber each other. It does require a separate RAM, however, at locations D000-D07F for strictly monitor use. In return, it keeps HUMBUG storage strictly compatible with any stack or storage assignment made by other programs, so there is never a problem.

In my system, the storage at D000 is provided by the 4K board I mentioned earlier. In two other systems that are currently running under HUMBUG, the memory is provided by the CPU board's 6810, relocated to C000-DFFF as also mentioned last month.

I/O Control from the Keyboard

HUMBUG's control terminal is a serial terminal using an MP-S card at port 1, which provides all input to the monitor,

and also standard output. Location PORECH (A00A) contains $8004, which points to this port. By changing this number, you can redirect the control port to an MP-S card at any other port. (I'm describing the common version of HUMBUG; my own has its I/O at $F804.)

In addition, HUMBUG can provide an output to a second MP-S at port 0, to a user-written output routine in another EPROM or (in the 3K 2708 version) to the Percom video board.

Any time that the monitor is looking for commands or any time that INEEE or OUTEEE is called, HUMBUG checks this port for a control-S break character arriving from the keyboard. When a control-S is detected, HUMBUG echoes with a bell (control-G) and halts all current I/O.

When I/O is halted, HUMBUG waits for one more character, which is used for controlling monitor ports. If it is received by INEEE, then it is not returned back to whatever program called INEEE. This provides control of output ports without upsetting other programs. This control character can be one of the following:

CR—cancels the current program and usually does a return to the monitor. But the return is handled through a pointer in RAM, so that other programs could change the pointer and force a return to themselves.

0—turns port 0 on and off.

1—does the same for port 1.

D—does the same for a user-written port routine.

P—turns the pause feature on and off. When the pause feature is on, output will stop every 16 lines to allow it to be read when using CRT terminals.

Any other character is ignored. The 0,1 and D characters toggle their corresponding ports; if a port is on then it goes off, if it is off then it goes on. Since these characters are not echoed or even returned to calling programs, ports can be turned on and off in the middle of input or output.

The video board output normally runs all the time and is not controlled. (There is a flag in monitor RAM, however, that disables it if I want to use it for graphics or memory-mapped output.)

When another port is on, then the video output simply runs at the speed of the slowest port. But when all other ports are turned off, then the video board runs at breakneck speed, limited only by CPU speed.

This feature is extremely versatile. Not only does it allow precise printer control, but it also permits rapid skipping ahead at video speed. (The 2K 2716 version, which does not support the video board, will skip ahead even faster when you turn off all output.) Moreover, the control-S/CR combination allows you to abort jammed programs without reaching for the RESET button.

Extended Debugging Facilities

My third requirement for improved debugging power was met in several ways. First, the HD command allows a hex dump of selected memory areas. The DE command prints a "desembled" listing of machine code, formatted by address and instruction. Thus, a DE dump of a program might go like this:

```
1000 86 41
1002 BD E1D1
1005 4C
1006 B7 1103
```

An AO command outputs memory data in ASCII so I can scan for strings. An FM command allows filling memory with a specified byte. This is convenient to fill memory with 3F (SWI) instructions to catch programs that go wild. The FI command allows searching memory to find one, two or three bytes. The MO command moves memory contents from one place to another, even if the new area overlaps the old area.

But the most important function is the breakpoint and single-step facility. Up to four breakpoints can be set in programs, and whenever a program encounters such a breakpoint (or any SWI instruction anywhere), an interrupt returns control to HUMBUG, which then prints out the register contents

and stops. HUMBUG keeps track of breakpoint locations and the instructions existing in those locations and prints a listing of them whenever the BP (breakpoint print) command is given. This reminds you where you have put the breaks. An important feature is that HUMBUG doesn't forget about them either when a jump back to the monitor is done, or when RESET is pressed.

SS is used for single-stepping through programs. Each time you type SS, HUMBUG prints out the address and code of the next instruction, executes it and then prints out the contents of all registers after the instruction is completed. It will single-step all instructions except WAI, SWI and RTI, and cannot single-step into or through ROM. HUMBUG prints out NO! whenever any of these are attempted.

FCROM

FCROM occupies addresses FC00-FFFF. It contains the reset and interrupt vectors that the 6800 CPU needs at locations FFF8-FFFF. So, without this ROM, the system cannot function at all.

FCROM contains all of the common MIKBUG I/O routines. But since this ROM is at the end of memory, none of these routines are at MIKBUG-compatible addresses. Instead, they are simply consecutively placed wherever they fit. To allow future changes, though, they are vectored through a jump table that starts at FC00:

```
FC00 JMP COLDST
FC03 JMP WARMST
FC06 JMP HOTST
FC09 JMP INEEE
FC0C JMP OUTEEE
etc.
```

Even when FCROM is changed in the future, these pointers will stay in the same place, and so external jumps into FCROM will stay unchanged.

OUTEEE and INEEE provide all of the port control features mentioned before. In addition, FCROM has a command processor that accepts monitor commands from the keyboard and processes them. But it only recognizes two commands—ME for memory examine and change and JU to jump to a user program. These are the absolute minimum that the monitor could have and still work.

Monitor Extendability

My fourth major requirement was to allow the monitor to be changed or expanded without too much work. As it now stands, I can add EPROMs without changing the existing ones. Moreover, I can even unplug some of the existing EPROMs from the system, and the rest of the monitor will still work! (Since the 2716 version consists of just one EPROM, this obviously doesn't apply to it.)

The 2708 version of HUMBUG consists of three 1K EPROMs: FCROM, E0ROM and E4ROM.

FCROM is completely self-contained and will run all by itself, even when the other EPROMs are unplugged. It contains all port control and video board control and, with the ME and JU commands, can load and execute other programs. But it is obviously limited; it relies on the other EPROMs in the system. It also doesn't have MIKBUG-compatible entry points, although it does have all the required routines.

This is where the extendability feature comes in. Notice in the above table that there is an entry point at FC00:

```
FC00 JMP COLDST
```

This is the main entry point when you first turn the system on or when you push RESET. This is a "cold-start," which initializes ports 0 and 1 and initializes the video board.

Once this is done, the FCROM program checks to see whether there is a ROM starting at address E000. If there isn't, then it proceeds with a "warm-start" initialization, where the program turns on port 1, turns off other ports and sets more registers. But if it detects that there is a ROM at E000, it executes a JSR to that ROM before doing the warm-start. This gives E0ROM a chance to execute a cold-start too.

When E0ROM is finished with

```
                    * INTERRUPT VECTORS
FFE8 FE A000  IRQV   LDX   IRQ         IRQ VECTOR VIA A000
FFEB 6E 00           JMP   0,X
FFED FE A012  SWIV   LDX   SWIJMP      SWI VECTOR VIA A012
FFF0 6E 00           JMP   0,X
FFF2 FE A006  NMIV   LDX   NMI         NMI VECTOR VIA A006
FFF5 6E 00           JMP   0,X
   (FFF8)            ORG   $FFF8
FFF8 FF E8           FDB   IRQV IRQ VECTOR
FFFA FF ED           FDB   SWIV SOFTWARE INTERRUPT
FFFC FF F2           FDB   NMIV NMI VECTOR
FFFE FC 00           FDB   COLDV RESTART VECTOR FOR RESET
```

Listing 1. Interrupt and Reset vectors.

its cold-start, it checks for the presence of a ROM at either E400 or E800; if it detects one, it jumps there. Each EPROM gets its chance at a cold-start initialization. If E4ROM is installed at E400, it gets control; if not, then control either goes to the next ROM (if any) or returns to FCROM. Initialization is divided into cold-start and warm-start, and each of these transfers control from ROM to ROM.

When all initialization is completed, FCROM takes over again and looks for a command. If an ME or JU command is entered, then FCROM executes a memory change or jump itself. Otherwise, it puts the two command characters into accumulators A and B and transfers control to other ROMs, in turn. If one of these recognizes a valid command, it executes it; otherwise, control goes to the next ROM. Ultimately, control passes back to FCROM.

Passing control back and forth between ROMs allows more ROMs to be added at any time. Moreover, if one ROM is unplugged, the remaining ROMs still get control and can still execute their own commands. In this way, you can expand or modify HUMBUG without reburning all three EPROMs. But there is a price to be paid: an additional amount of housekeeping in each EPROM, which takes up about 40 bytes.

E0ROM

E0ROM, the second 2708, is at locations E000-E3FF. Although the system will run with just FCROM, E0ROM is essential for MIKBUG compatibility because the E0ROM has sixteen MIKBUG-compatible jump vectors that point to the corresponding locations of FCROM. For instance, location E1AC of E0ROM contains an instruction that says JMP to $FC09, which is the actual entry point for INEEE in FCROM. Each MIKBUG entry point has such a JMP.

This is a different approach from SWTBUG and other monitors, which simply put these routines at the same addresses as MIKBUG did and then try to fit everything in. Here all the routines are elsewhere, and only JMP instructions exist.

```
                    * JUMP VECTORS
   (FC00)                  ORG    $FC00
FC00 7E FC33  COLDV  JMP   COLDST    COLD START ENTRY POINT
FC03 7E FC52  WARMV  JMP   WARMST    WARM START ENTRY POINT
FC06 7E FC94  HOTV   JMP   HOTST     HOT START ENTRY POINT
FC09 7E FD93  INEEEV JMP   INEEE     INPUT CHARACTER ROUTINE
FC0C 7E FDFD  OUTEEV JMP   OUTEEE    OUTPUT CHARACTER ROUTINE
FC0F 7E FD79  CRLFV  JMP   PCRLF     PRINT CR/LF
FC12 7E FD85  PDATAV JMP   PDATA     PRINT A STRING
FC15 7E FDED  INCH8V JMP   INCH8
FC18 7E FD49  INHEXV JMP   INHEX
FC1B 7E FD24  BYTEV  JMP   BYTE
FC1E 7E FD16  BADDRV JMP   BADDR
FC21 7E FD5E  OUT2HV JMP   OUT2H
FC24 7E FD36  OUTHLV JMP   OUTHL
FC27 7E FD3A  OUTHRV JMP   OUTHR
FC2A 7E FD69  OUT2SV JMP   OUT2HS
FC2D 7E FD67  OUT4SV JMP   OUT4HS
FC30 7E FD6B  OUTSV  JMP   OUTS
```

Listing 2. FCROM jump table.

Woven in between these JMPs are the cold- and warm-start routines, the command processor that recognizes monitor commands and routines for the following commands:

LO—Load MIKBUG-formatted tape
PU—Punch/Save MIKBUG-formatted tape
EN—Punch end-of-tape with program counter and S9 code
FD—Bootstrap for Flex disk
PD—Bootstrap for Percom disk (or go to C000 EPROM)
GO—Go to user program using A048/A049 address
CL—Clear terminal screen
FI—Find one, two or three bytes in memory
HD—Formatted hex dump of memory
FM—Fill memory with a byte
CS—Compute a 16-bit checksum of memory contents
MT—Perform a memory test
PC—Print contents of A048/A049 (program counter).

Thus, this ROM puts in all of the necessary routines to make HUMBUG compatible with user programs and also puts in all the common SWTBUG commands, except breakpoints and register examine.

E0ROM has one more routine—FROMTO. This routine essentially asks for an input from the keyboard of a FROM address and a TO address, which are placed into BEGA location A002 and ENDA location A004 in the monitor scratchpad. This routine is called by most other commands to specify the beginning and end of desired memory.

The PU command is a good example. In SWTBUG or MIKBUG, locations A002 and A004 had to be preset to the starting and ending locations before calling the P command. In HUMBUG, the PU command uses the FROMTO routine to ask for the beginning and ending locations. This routine is set up so that entry of a carriage return will make it use the previous values.

E4ROM

E4ROM is the third 2708 and occupies addresses E400 through E7FF. Its cold-start and warm-start initialization and passing control to the next ROM are similar to those of E0ROM, but E4ROM adds the following commands:

DE—Desemble memory and print machine-language codes
BP—Breakpoint printout
BR—Breakpoint set or reset
CO—Continue after a breakpoint
RE—Register examine
SS—Single-step
AI—ASCII input into memory
AO—ASCII output from memory
MO—Move memory contents

The exact functions of these will become clear when we examine the actual programs.

Since the system is set up to allow more ROMs to be easily added, there are obviously others available. E8ROM, for instance, adds commands to compare memory contents, change terminal baud rate from the keyboard and change control ports. But these are just frosting on the cake, not really needed for most systems.

Let's examine some of the actual HUMBUG code.

Initialization and Reset

A 6800 requires four address vectors to be located in the top eight memory locations, FFF8 through FFFF, which are used to vector resets and interrupts. These four vectors are:

FFF8 and FFF9—IRQ vector
FFFA and FFFB—SWI vector
FFFC and FFFD—NMI vector
FFFE and FFFF—Reset vector

When you press the reset button or when an interrupt occurs, the 6800 pulls the appropriate address out of one of these four locations and puts it in the program counter. This causes a jump to that address. For that reason, when the system is first turned on, at least the reset vector and the routine it points to must already be in memory. This is why every 6800 system has its ROM located at the very top of memory.

Listing 1 shows the portion of HUMBUG's FCROM that contains the very top of memory. FFF8 through FFFF contain these four vectors: IRQ points to FFE8, SWI points to FFED, NMI points to FFF2 and reset points to FC00. Thus, when a reset is completed, the 6800 starts executing from location FC00, which is the beginning of FCROM.

The interrupt vectors all point to locations in ROM, shown just above that. When an interrupt occurs, the computer goes to the appropriate routine, loads a number from RAM into the index register and then does an in-

Basic Software

Put your computer to work doing something useful. Now *two* books, **Practical Basic Programs** and **Some Common Basic Programs**, offer a total of 116 practical, *useful* programs.

The programs cover a variety of practical applications: finance, management decision, statistics,mathematics and science. Each book contains complete program listings in a general version of Basic that will work on most popular computers. You can use any of the programs *even if you don't know how to program — just copy it from the book into your computer and go!*

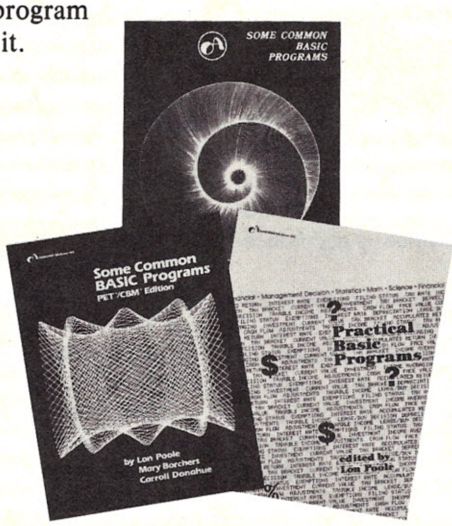
Documentation. Always a must, Osborne offers complete documentation with each program. It includes discussion of what each program does, how it works, and how to use it. An example of each program in action shows you exactly how the program will look when you use it.

SOME COMMON BASIC PROGRAMS. Now in its third printing, more than 50,000 readers have found useful programs here. It contains 76 programs, including Linear Programming, Tax Depreciation Schedule, Linear Interpolation and Coordinate Plot.

Commodore PET and CBM users. You can purchase a special edition of this book, just for use with the Commodore computers. All 76 programs have been modified to run optimally on the PET and CBM, so you won't have to waste a minute adapting them yourself. Be sure to order Some Common Basic Programs - *PET/CBM Edition.*

To go with the PET/CBM edition, you can also purchase the programs ready-to-run on cassette or diskette.

Radio Shack TRS-80 users. You too can purchase all 76 programs adapted specifically to your computer with our TRS-80 cassette. Use the standard edition of Some Common Basic Programs for documentation.



PRACTICAL BASIC PROGRAMS. Just published, this collection of 40 programs covers slightly more sophisticated applications than Some Common Basic Programs. There are extra practice problems, with answers, to help you fully understand what these programs can do. Program titles include Income Averaging, Lease/Buy Decision, Syndicated Investment Analysis, Home Budgeting, Critical Path Method, Nonlinear Breakeven Analysis, Statistical Estimation Theory, and Lagrangian Interpolation.

Osborne/McGraw ✓149
means documented software

Some Common Basic Programs (book)	# 06-3	$12.50
Some Common Basic Programs - PET/CBM ed. (book)	# 40-3	$12.50
Some Common Basic Programs - PET/CBM cassette	# 25-X	$15.00
Some Common Basic Programs - PET/CBM diskette	# 33-0	$22.50
Some Common Basic Programs - TRS-80 diskette	# 32-2	$15.00
Practical Basic Programs (book)	# 38-1	$15.00

To order, return coupon with check or money order. Include 75¢ per item for 4th class mail, $1.25 per book UPS, or $2.50 per book air mail in the U.S.

California residents also include local sales tax. To place an order by phone call 415/548-2805.

Book	Price	Quantity	Amount

Name ____________ Tax [ ]
Address ____________ Shipping [ ]
City ________ State ____ ZIP ____
Phone ________ HOW TO SHIP ________ TOTAL [ ]

OSBORNE/McGraw-Hill
630 Bancroft Way Dept. K5
Berkeley, CA 94710

```
              * COLDSTART INITIALIZATION
FC33 8E D07F  COLDST LDS   #$D07F     SET STACK TO MONITOR AREA

              * INITIALIZE I/O PORTS
FC36 CE 8000         LDX   #$8000
FC39 86 03           LDA A #3         PORT 0 AND 1 ACIA
FC3B A7 00           STA A 0,X
FC3D A7 04           STA A 4,X
FC3F 86 0B           LDA A #11
FC41 A7 00           STA A 0,X
FC43 A7 04           STA A 4,X
FC45 BD FFA5         JSR   VINIT      INITIALIZE VIDEO

              * SEE IF OTHER ROMS REQUIRE COLD START INITIALIZATION
FC48 B6 E000         LDA A $E000      CHECK ROM-E0
FC4B 81 7E           CMP A #$7E       IS THERE A JUMP?
FC4D 26 03           BNE   WARMST     NO
FC4F BD E000         JSR   $E000      YES, GO TO IT
```

Listing 3. FCROM cold-start initialization.

dexed jump to the address given in the index register. This address is actually specified through RAM and can therefore be changed by user programs, even though the JMP instructions themselves are in ROM. The three addresses used are exactly compatible with SWTBUG:

IRQ is A000
SWI is A012
NMI is A006

FCROM Jump Table

FCROM contains routines that are subject to future change. To avoid having to change other software, all these are handled through a jump table (sometimes also called a "transfer vector") as shown in Listing 2. In particular, note that FC00 is the start location to which the computer jumps on a reset. This is called COLDV (cold-start vector), and it jumps to COLDST at FC33. Two other entry points are WARMV and HOTV, followed by vectors or pointers to all the MIKBUG-compatible routines.

Cold Start

Listing 3 shows what happens at a reset (or cold-start; a jump to E0D0, which is the MIKBUG/SWTBUG reset address, also winds up at this location).

First, the stack pointer is set to point to the monitor stack at D07F. Then, MP-S ACIAs on ports 0 and 1 are reset and then initialized, followed by a jump to the video board initialization routine. In the case of HUMBUG, this is exactly the same as Percom's suggested video driver initialization, and so there is no need to show it here. If you have this video board, you already have a listing of it; if you don't, then you don't need it and can replace it with initialization for another video board or skip it.

The last four lines of cold-start check to see whether there is another ROM at address E000. Since all HUMBUG ROMs start with a jump table, we check to see whether there is a 7E or JMP instruction at address E000. If not, we continue to WARMST. If there is a JMP, we execute a JSR to E000.

Cold-Start of Other ROMs

As it turns out, E0ROM doesn't need any cold-start initialization. Unfortunately, the overhead involved with the expandability of HUMBUG requires that we go through some testing to check for a following ROM (see Listing 4). Here we see the JMP at location E000, which leads to CINIT. Since E0ROM has many MIKBUG-compatible jumps, a lot of its routines have to be squeezed between these jumps. In this case, the CINIT cold-start initialization is placed right after the INEEE vector at E1AC, which is also shown in Listing 4.

The NOP at CINIT shows where the initialization would go, if there was some. The following steps check for a JMP at the start of the next ROM at address E400 and jump to it if present. If not, then they check for a JMP at the start of a ROM at E800 and again jump to it if present. If neither is present, then there occurs an RTS, which returns back to FCROM's warm-start procedure.

These steps check for a JMP both at address E400 and at E800, so that if an additional ROM is installed at E800, but the one at E400 is pulled out, then the system will simply skip past the removed ROM. The purpose is to allow the monitor to function at least partially, even if some of its ROMs are pulled out. The only crucial ROMs are FCROM and E0ROM, although the system will work even with just FCROM.

Although E0ROM doesn't need cold-start initialization, E4ROM does. Its cold-start initialization is shown in Listing 5. Notice that E4ROM tries to differentiate between a reset or jump to the cold-start location E0D0, as opposed to a real cold-start right after the first power-on. The reason is because breakpoints have to be handled differently.

When you first turn on the power, the list of breakpoints maintained by HUMBUG has to be erased so that, if any new breakpoints are established, HUMBUG doesn't accidentally

```
              * E0 ROM ENTRY VECTORS

E000 7E E1AF  CINITV JMP   CINIT      COLD START INITIALIZATION

E1AC 7E FC09  INEEEV JMP   INEEE      VECTOR TO FC ROM

              * CINIT - COLD START INITIALIZATION

E1AF 01       CINIT  NOP              NONE REQUIRED FOR THIS ROM
              * SEE IF OTHER ROMS REQUIRE INITIALIZATION
E1B0 B6 E400         LDA A $E400      CHECK NEXT ROM
E1B3 81 7E           CMP A #$7E       IS THERE A JUMP?
E1B5 27 08           BEQ   CMORE4
E1B7 B6 E800         LDA A $E800      CHECK THE ROM AFTER THAT
E1BA 81 7E           CMP A #$7E       IS THERE A JUMP?
E1BC 27 04           BEQ   CMORE8
E1BE 39              RTS              NO, RETURN TO FCROM
E1BF 7E E400  CMORE4 JMP   $E400      YES, GO INITIALIZE
E1C2 7E E800  CMORE8 JMP   $E800      YES, GO INITIALIZE SECOND ROM
```

Listing 4. E0ROM cold-start initialization.

```
              * E4 ROM ENTRY VECTORS

E400 7E E409  CINITV JMP   CINIT      COLD START INITIALIZATION
              * CINIT - COLD START INITIALIZATION

              * CHECK WHETHER THIS IS POWERUP OR RESET OF SYSTEM
E409 CE 1234  CINIT  LDX   #$1234     CHECK POWER-UP LOCATIONS
E40C BC D028         CPX   POWUP
E40F 26 05           BNE   PUP
E411 BC D02A         CPX   POWUP+2
E414 27 1C           BEQ   RESET

              * INITIAL POWER UP SEQUENCE
E416 CE E6BF  PUP    LDX   #BKRETN    INITIALIZE BREAKPOINT ISS ADDRESS
E419 FF A012         STX   SWIJMP
E41C CE 1234         LDX   #$1234
E41F FF D028         STX   POWUP
E422 FF D02A         STX   POWUP+2    INITIALIZE POWUP FLAGS
E425 CE D036         LDX   #BKTAB
E428 86 FF           LDA A #$FF
E42A C6 0C           LDA B #12
E42C A7 00    BKERAS STA A 0,X        ERASE BREAKPOINT TABLE
E42E 08              INX
E42F 5A              DEC B
E430 26 FA           BNE   BKERAS     REPEAT IF NOT FINISHED

              * SEE IF OTHER ROMS REQUIRE INITIALIZATION
E432 B6 E800  RESET  LDA A $E800      CHECK NEXT ROM
E435 81 7E           CMP A #$7E       IS THERE A JUMP?
E437 27 08           BEQ   CMORE4
E439 B6 EC00         LDA A $EC00      CHECK THE ROM AFTER THAT
E43C 81 7E           CMP A #$7E       IS THERE A JUMP?
E43E 27 04           BEQ   CMORE8
E440 39              RTS              NO, RETURN TO FCROM
E441 7E E800  CMORE4 JMP   $E800      YES, GO INITIALIZE
E444 7E EC00  CMORE8 JMP   $EC00      YES, GO INITIALIZE SECOND ROM
```

Listing 5. E4ROM cold-start initialization.

clobber a program by restoring what it thinks is a prior breakpoint.

On the other hand, when you press the reset button or make a jump to the cold-start location E0D0 (or FC00), you don't want to erase the breakpoint table because doing so would make you lose track of locations that have been replaced by a break. So we need a way of telling the difference between the two kinds of resets.

For this reason, four locations in monitor RAM, called POWUP, and located at D028 through D02B, are used as a flag. When you first turn on the power, these locations will contain some random numbers. CINIT in E4ROM (Listing 5) checks the contents of these locations. If the contents are 12, 34, 12 and 34, respectively, then the program assumes that this is not a real cold-start, and so a jump is made to RESET. But at the first cold-start, these locations will be random and will therefore not contain this particular combination. (The chance of their just accidentally holding this number at power-up is about 1 in 4 billion!)

In that case, the routine at PUP will be performed. This initializes the address for an SWI to the return address BKRETN used for breaks, places the 12-34-12-34 combination in POWUP and erases the breakpoint table BKTAB. Then it goes to RESET. (Once POWUP is set to 12-34-12-34, all subsequent resets will skip this segment.)

The final part of the cold-start procedure again checks whether there are other ROMs, this time at E800 and EC00, and jumps to them if present. Otherwise, an RTS brings us back to FCROM, which will continue with the warm-start initialization. Remember that FCROM went to E0ROM with a JSR. Each ROM then continued to the next ROM with a plain JMP, so that an RTS will bring us all the way back to the first JSR in FCROM.

Next month we'll conclude the listing of this "Monitor to End All Monitors." ■

Dial-up Directory

This Directory tells you how to live with an overactive dialing digit.

Frank J. Derfler, Jr.
PO Box 691
Herndon, VA 22070

How much is your phone bill? Most of the letters and messages I receive as a result of these articles start with "I will hate to see my phone bill, but I sure am having a good time . . . ."

This article introduces one service that may make a difference in the size of your monthly donation to the telephone company and examines a creative bulletin board program. Finally, I will attempt to throw some light into the dark corner of TRS-80 telecommunications.

Sprint

Is your long-distance phone bill over $50 a month? Do you live in or near one of the top 50 or so cities in population? Do you usually call to large population centers? Do you have a push-button phone?

If you answered yes to all the above, you may want to "sprint" to Southern Pacific Communications (SPC) to save a few dollars. Sprint is the name of SPC's long-distance dial telephone network. It picks you up from your local telephone system, carries you across country and deposits you in the local system of many large population centers across the country. The service will carry voice or voice bandwidth data (standard 300 baud is fine).

SPC has been in business since 1970. Today, they operate more miles of intercity microwave circuitry than any other specialized common carrier. The number of specialized carriers such as SPC is growing, and they are all expanding the areas they serve. SPC projects continued growth to more areas of the country. Since 50 percent of the U.S. population lives in the 40 largest cities, SPC is able to serve most of us right now.

To use Sprint, you first dial a local number on your standard push-button instrument. This local number connects you with the Sprint network. You then dial the long-distance number you want and your Sprint authorization code. The network takes over, and you proceed as if you have made a regular voice or data call.

The cost depends upon when and how long you call. If you compare "prime time" 8 AM to 5 PM calls, Sprint saves around 30 percent. Sprint drops to low rates at 5 PM, so if you compare their calls with the regular evening rate, you will probably save more.

From Alabama, a 15-minute call to Los Angeles at 6 PM costs me $3.69 with normal telephone service. The Sprint call runs about 11 cents per minute (varies with distance), plus a ten cent termination fee, for a total of $1.75. You have to also prorate a $10 monthly subscription cost over the number of calls you make.

For example, if you add a pro rata charge of 25 cents to total $2, you have almost saved enough on that one call to buy a 21L02 static RAM for your memory board. (You could add a pro rata share of the regular telephone service and installation costs to the $3.69, but I will treat them as sunk costs.) Sprint is billed in six-second increments. The standard telephone service rounds up in all cases.

Since most of us use our dial-up communications at night and during the weekend, let's consider the cost of a 6 AM call. On the regular phone system, it would cost $2.30, so you would only save a quarter or so with Sprint. The saving is less, but it is also less convenient.

Sprint is not for everyone. It doesn't serve the Northwest or Northcentral states. There is a minimum $25 a month charge, so you have to make quite a few calls. You have to analyze your own situation and then determine if the network has a local connection point near you before you decide. If you are interested, contact: SP Communications, PO Box 974, Burlingame, CA 94010, (415) 692-5600.

System Spotlight

Anyone interested in the finer points of bulletin-board or message services should dial into the system in Endicott, New York, (607) 754-5571. Bob Iannucci has put together this service with an eye toward both economy and individual attention. The entire program runs on one single-sided single-density, small, North Star floppy disk. The program is in 8080 assembly language, so there is no overhead for CP/M or BASIC. The files are self-maintaining, and no intermediate or other records are needed.

This flexible system assumes you know what you are doing until you prove you don't. The prompt commands are short to save time for the experienced, but if you need help, detailed explanations are available at every level.

Bob's system asks new users to provide a four-letter password and a user ID. The password is used to kill or edit your old messages. Other people use the ID to send you messages, which you can quickly recall with a special summary command.

The ID prevents users from being inconsistent with the names they use to sign on and send messages. Other systems have a similar user name feature, but if I sign on as John Doe and the message is sent to Mr. Doe, I will not see the message on a sort by name.

Bob must provide an ID to user name cross-reference, and use of this file takes time, so it still isn't a perfect solution. However, regular correspondents will quickly learn each others' IDs.

The user ID also provides semipermanent personalization of the system. The first time you sign on, you must tell the system the number of nulls you need, if you have a bell, what your keyboard delete code is, if you can use lowercase and if you want an echo. The system remembers these factors when you use your ID in the future. This is a time-saver.

This system, like several others, now allows "stacking" of commands. If you know what questions are coming, you can answer them in advance by separating the answers with semicolons. Without the stacking feature, the sequence appears as:

```
IS THIS YOUR FIRST TIME ON THIS SYSTEM? no
HAVE YOU USED A BULLETIN BOARD SYSTEM
BEFORE? yes
WHAT IS YOUR NAME? john doe
```

You could sign on as follows:

```
IS THIS YOUR FIRST TIME ON THIS SYSTEM?
n;y;john doe
```

Stacking is an obvious time-saver if you are familiar with the system. Always check the help (H or ?) command for information on how to make bulletin board systems really work for you. Bob is not in the business of selling software and his program is machine specific, but the ideas and style he has used can be copied by all.

List of part-time operations run by individuals and computer stores and full-time systems run by commercial companies for communications with their users. These phone numbers were confirmed from my list of 160 reported listings; I have been on each one of these systems.

LOCATION	PHONE NO.	COMMENTS
California		
Berkeley	415-527-0400	Proxima. The official message system of North Star Computer Company.
Brentwood	213-395-1592	ABBS. 4 PM–9 AM and weekends.
Santa Clara	408-296-5799	CBBS-type software running on a PDP-8 minicomputer with a hard disk.
Iowa		
Dubuque	319-557-9618	ABBS.
Kentucky		
Louisville	502-245-8288	ABBS. 8 PM–6 AM.
New Jersey		
Wycoff	201-891-7441	ABBS. 10 PM–6 AM.
Tennessee		
Memphis	901-362-2222	TRS-80-based system. Hobbyist Forum, slanted toward local users.
Texas		
College Station	713-693-8080	ABBS.
Washington		
Seattle	206-244-5438	Apple Crate II. Features the "Apple Doctor" service.
Seattle	206-723-3282 (206-723-DATA)	TRS-80-based system serving a small but active company called The Peripheral People. Ask about their "Micro-connection" modem.

TRS-80

The world of TRS-80 communications has been a cloudy one. National advertising of non-Radio Shack products for communications is limited, and model numbers and types of software have been a subject of some confusion. I have been waiting to get some solid information before making any comments in this series about TRS-80 systems. Now I can discuss one software package that provides good communications capability for the TRS-80. But first, I'll examine the hardware.

If you are going to communicate using a TRS-80, you need the expansion interface, the RS-232 card and a modem. Some RS-232 hardware that does not use the expansion interface is being advertised (see "TRS-80 Serial I/O for Less" in the April 1980 *Microcomputing*, p. 100). If you want to use a smart terminal program with one of these boards, you'd better be able to write your own software. (Note: that is not true of the "Micro-connection" recently announced for the TRS-80, which we will review soon.)

The Radio Shack RS-232 cards apparently got off to a rough start because of some bad quality control, and they still give problems in physical connection. If you have memory errors, system reboots, random error messages and other seemingly unrelated problems, check the connection between the RS-232 card and the expansion interface.

Any of the RS-232-type 103 modems will work with the TRS-80. The Novation CAT modem, sold under the Radio Shack label, does a fine job. Disk systems are not necessary, but they sure are helpful. The Term program sold by Radio Shack is a dumb terminal program that will run without a disk. Other software has more capabilities.

The Bottom Shelf, Inc., in Atlanta has come on strong in the TRS-80 software market with a Terminal Control Program (TCP) written by Barry Mulligan that is clever in concept because it is both a stand-alone package and a utility or subroutine to be called as a part of larger programs. The package allows communication with any other computer or terminal and the transfer of programs, data and memory blocks.

All of the RS-232 parameters (word length, etc.) are controlled from the keyboard. Versions for 16K, 32K and 48K are provided, and the package will work from tape or disk. Saving data is done by the standard tape or disk operating system.

Several levels of operation can be selected. Dumb terminal operation is available, but features such as control codes, error checking and a bell (audio tone) are provided for the dumb terminal. The TCP manual calls this KSR—after the keyboard send receive designation often used with printing terminals on communications circuits.

The TCP software also allows you to transmit and receive BASIC programs. The compressed format used internally by the TRS-80 is transmitted under some options. This saves long-distance phone-call time, but you must have another TRS-80 at the other end in this mode.

A clever option allows the TCP software to interact with running programs. A program can run for a while, call TCP, exchange data over a modem, terminate TCP and continue running. A BASIC program is included to read ASCII files (such as Electric Pencil), call TCP and transmit or receive messages. This capability is similar to the ASCII Express program for the Apple II.

The manual that Barry has written is aimed at the practical user. Each option is explained in detail, and good step-by-step instructions are provided. This program is easy to use and well documented. It should serve the needs of both expert and novice keyboard pounders. The program costs about $20.

Do you market products for microcomputer telecommunications? Do you operate a message system? Let me know what you are up to. Problems? Ask me. Send paper mail to PO Box 691, Herndon, VA 22070 (include a stamped envelope if you want a reply). Save a tree and send electronic mail via the Remote Northstar (404) 939-1520, or to TCB967 on The Source. ■

Let PET Design Your Next Power Supply

Make your computer an engineer/draftsman.

William R. Moore
2035 Barberry Lane
Bowling Green, KY 42101

One of the first programs I purchased after getting my PET computer was George Duisman's Bridge Challenger program. Since the program was almost 8K in length, my PET ran out of memory after playing only a few hands.

I decided to add an additional 4K of memory to the PET, since the memory expansion connector made this addition easy. However, I soon found out that the PET power supply was not capable of supplying the additional current. A 5 volt, 1.2 Ampere power supply would be needed to power my memory board.

As I pondered over the design, I decided to let the PET design the power supply. After all, a computer should be capable of doing basic power-supply design, and maybe the PET graphics could be used to draw the schematic.

I wrote the program to use the National Semiconductor LM317 regulator which is an adjustable three-terminal regulator capable of providing dc voltages from 1.2 to 30 volts at currents up to 1 Ampere. The LM317 comes in three case sizes—TO-5, TO-220 and TO-3 for increased power dissipation. Since the LM317 can only handle currents up to 1 Ampere, larger currents will pass transistors.

The Program

The program will design a dc power supply from 1.2 volts to

Program listing.

```
1   GO TO 9Ø
2   REM GRAPHICS FOR SCHEMATICS
4   PRINT"
5   PRINT"0####P___________IN_0#####P_OUT
6   PRINT"%____'________U@@2@@%_LM317_'##P###W
7   PRINT"%____'##P##PD3]__'__%_____'___/
8   PRINT"%120V'D1:$_:$_]__]__L$$$$$:__\
9   PRINT"%PRI_'__/\_\/_]__'_____'______/"R1"Ø"
1Ø  PRINT"%____'__P#_P#_]__:$+___'_____\
11  PRINT"%____'__'@1'@@K_____C__'$$$$$$:
12  PRINT"%____'__:$]:$____P#____'
13  IF V>=10 THEN GO TO 15
14  PRINT"%"V"_'D2\/]/\D4__'_____/":GO TO 16
15  PRINT"%"V_"'D2\ /]/\D4__'_____/
16  PRINT"%SEC_'__P#]P#____'_____\"R2"Ø"
17  PRINT"%____'######_____'_____/
18  PRINT'L$$$$:____]______'_____\
19  PRINT"__________J@@@@@@1@@@@@1@@@@@@@@@W
2Ø  PRINT"_________________D1-D4=1N4ØØ2
21  RETURN
22  PRINT"
23  PRINT"0####P_D1______IN'######POUT
24  PRINT"%____'__'/%______'_LM317'$$$$$$W
25  PRINT"%120V'##P/0P##P##P______'__'
26  PRINT"%PRI_'_____'__'__'$$$$$$__/
27  PRINT"%____'_____'__'___ADJ.%____\"R1_"Ø
28  PRINT"%____'@@.__'__:$+_____%____/
29  PRINT"%"V_"'__)__'____C_____%____\
3Ø  PRINT"%SEC_'___D2'__#0______L$$$$:
31  PRINT"%C.T.'__'\L:___%______%
32  PRINT"%____'##P/%____%______\
33  PRINT"L$$$$:__________%______/"R2__"Ø
34  PRINT"__________Ø@@@@%______\
35  PRINT"__________)____%______/
36  PRINT"_D1,D2=1N4ØØ2__L$$$$$$L$$$$$$$$W
38  RETURN
4Ø  PRINT"cs___________________$2N3Ø55("X")"
41  PRINT"_________0#######\_V0######P
42  PRINT"_________%_______#P##______'
43  PRINT"__________%_2N29Ø7A'_5ØØØ___'
44  PRINT"_________0#####V%/##\/\/\/#P
45  PRINT"0#P______%_____#P##________'
46  PRINT"%_'_D1___%______'_0######P_'OUT
```

25 volts and load currents up to 5 Amperes. It will specify the necessary transformer characteristics for you, or you may specify an available transformer. If you specify the transformer, you must not call for a transformer significantly larger than the minimum required; otherwise, excessive power dissipation will occur in the power supply, especially for load currents greater than one Ampere.

Lines 5 through 62 in the program listing develop the power-supply schematics. Statements 5 through 62 use the hyphen for two distinctly different purposes. A hyphen above a character in the listing means that the shift key is to be depressed for that character. This is required since the majority of the graphic symbols on the PET keyboard are in the shift mode. The second use of the hyphen is to represent spaces when they are essential. Hyphens between characters (bottom of a line) are used for this purpose.

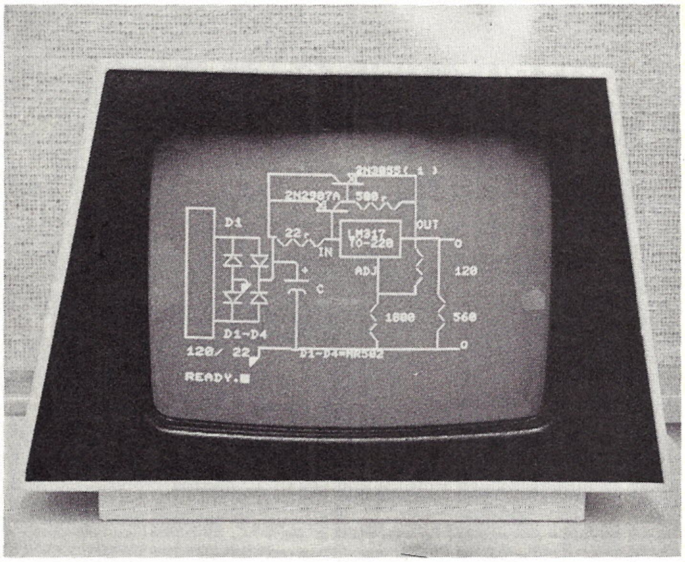


Photo 1. PET's graphics capabilities used for a 5 volt, .5 Amp power supply.

Statements 5-62 are double-spaced for clarity due to the double use of the hyphen. I have also used abbreviations used in the listing for clearing the screen (lowercase cs) and moving the cursor down (lowercase cd). I used the abbreviation for print (?) for all statements in the listing. Program execution may begin at statement 108 to eliminate introductory remarks.

Photo 1 shows the design for a 5 volt, half Amp regulated supply. A transformer having a minimum secondary voltage of 8 volts RMS would be required. The LM317 device contains most of the necessary electronics; only a 1400 uF, 12 V dc capacitor and two resistors are required to complete the design. The listing determines the maximum thermal resistance of the heat sink required.

The schematic diagrams generated by the computer for power-supply designs requiring one or more series pass transistors will always show one

```
47   PRINT"%_'______L_22Ø__'_%LM317_'$:$$$

48   PRINT"%_'#P##P_]\/\/\/##%TO-22Ø'_'___O#W

49   PRINT"%_'_:$_:$]_____IN_L######:__\__%"

5Ø   PRINT"%_'_/\_\/+@@I__________%____/__%

51   PRINT"%_'_#O_P#]__]+______ADJ%____\__%"R1"

52   PRINT"%_'_'@I'@K_@1@_________%____/__%

53   PRINT"%_'_$$):$__U2I_C_______L$$$$__%

54   PRINT"%_'\/_/\___]__________%______\

55   PRINT"%_'_P#_P#___]_________/________/

56   IF R2>=1000 THEN GO TO 58

57   PRINT"%_'_'__'____]_________\"R2_"___\"R3__:GO TO_59

58   PRINT"%_'_'__'____]_________\"_R2_"__\"R3"

59   PRINT"%_'#####____]_________/________/

6Ø   PRINT"L$:_D1-D4___]_________\________%

61   PRINT"________$$$$]$$$$$$$$$$L$$$$$$$L$$W

62   PRINT"120/"V_"%___#D1-D4=MR5Ø2

63   PRINT"_______)
64   RETURN
65   FOR J=Ø TO 4ØØ STEP 4Ø
66   POKE 33Ø92+J,32
67   POKE 33Ø93+J,32
68   NEXT J
69   POKE 33Ø92,99
7Ø   POKE 33Ø93,99
71   POKE 33251,64
72   POKE 33252,64
73   POKE 33253,64
74   POKE 33452,99
75   POKE 33453,199
76   GO TO 61Ø
9Ø   PRINT"cs THIS PROGRAM WILL DESIGN A"
91   PRINT"REGULATED D.C. POWER SUPPLY"
92   PRINT"USING A LM317 REGULATOR"
93   PRINT"cd THE MAXIMUM REGULATED D.C. VOLTAGE IS 25V"
94   PRINT"THE MINIMUM OUTPUT VOLTAGE IS 1.2 VOLTS"
95   PRINT"THE MAXIMUM OUTPUT CURRENT IS 5 AMPS"
96   PRINT"cd CURRENT LIMITING IS AUTOMATICALLY"
97   PRINT"PROVIDED BY THE LM317"
98   PRINT"cd THE_PROGRAM_WILL_SPECIFY______W.MOORE"
99   PRINT"A_TRANSFORMER_FOR_YOU_________1979"
1ØØ  PRINT"__________OR"
1Ø1  PRINT"YOU MAY SPECIFY A TRANSFORMER"
1Ø2  PRINT"cd ARE YOU READY TO RUN?"
1Ø3  PRINT"IF SO TYPE 1 AND RETURN"
1Ø4  PRINT"TO CONTINUE THE PROGRAM"
1Ø5  INPUT A
1Ø6  IF A=1 THEN GO TO 1Ø8
1Ø7  GO TO 1Ø5
1Ø8  PRINT"cs FINE, TYPE IN YOUR DESIRED"
1Ø9  PRINT"OUTPUT VOLTAGE"
11Ø  INPUT VOUT
115  IF VOUT>25 THEN GO TO 545
12Ø  IF VOUT<1.2 THEN GO TO 56Ø
125  PRINT"cd NOW TYPE IN YOUR DESIRED FULL"
13Ø  PRINT"LOAD CURRENT IN AMPERES"
135  INPUT IL
14Ø  IF IL>5 THEN GO TO 59Ø
145  IF IL<.Ø2 THEN GO TO 6Ø6
15Ø  VMIN=VOUT+3
155  EMAX=VMIN*1.3
16Ø  VRMS=1+INT (EMAX/1.4)
17Ø  PRINT"cd DO YOU DESIRE TO SPECIFY THE"
175  PRINT"TRANSFORMER FOR YOUR POWER SUPPLY?"
18Ø  PRINT"cd TYPE 1 FOR YES, Ø FOR NO"
185  INPUT SEL
19Ø  IF SEL<>1 THEN GO TO 24Ø
195  PRINT"cs DOES YOUR TRANSFORMER HAVE A CENTER TAP?"
196  PRINT"TYPE 1 FOR A C.T.,Ø FOR NO C.T"
197  INPUT CT
198  PRINT"NOW TYPE IN TOTAL RMS SECONDARY VOLTAGE"
199  INPUT VSEC
2ØØ  IF CT=Ø THEN GO TO 22Ø
2Ø5  VX=(VSEC/2)-VRMS
21Ø  IF VX<Ø THEN GO TO 22Ø
215  VRMS=VSEC/2
218  GO TO 235
22Ø  VX=VSEC-VRMS
225  IF VX<Ø THEN GO TO 6ØØ
23Ø  VRMS=VSEC
231  CT=Ø
235  EMAX=1.4*VRMS
24Ø  VDC=(EMAX+VMIN)/2
245  RL=VDC/IL:REM CALCULATE FILTER CAPACITOR
25Ø  C=(EMAX*1Ø↑6)/(2ØØ*RL*(EMAX-VMIN))
251  C=INT(1.2*C)
252  CVD=INT(1.25*EMAX)
255  D=VDC-VOUT
26Ø  PCAL=IL*D
265  R1=1ØØ
266  DIM DELTAR(39)
267  DIM RES(39)
268  DIM RS(2)
269  DELTAR(Ø)=4ØØØ
27Ø  REM INPUT RESISTOR VALUES
28Ø  DATA Ø,1Ø,15,22,33,47,56,68,82,91,1ØØ,12Ø,15Ø,18Ø,22Ø,
     27Ø,33Ø,39Ø,47Ø,56Ø
283  FOR L=1 TO 2Ø
285  READ RES(L)
287  NEXT L
29Ø  RES (21) = 68Ø
291  RES (22) = 75Ø
292  RES (23) = 82Ø
293  RES (24) = 91Ø
298  FOR I=1 TO 14
299  RES (24+I)=1Ø * RES (1Ø+I)
```

series pass transistor (2N3055) with the number of total paralleled transistors in parentheses. Photo 2 shows an example of a 20 V, 2 A power supply requiring one external 2N3055 series pass transistor.

In these cases, all of the load current will flow through the transistors, except for approximately 30 mA, which will flow through the LM317. This means that the regulator will not require a heat sink; only the series pass transistors will require a sink.

The power supply designed for my 4K memory expansion worked the first time and has continued to perform without problems. I built and checked out fifteen other power-supply designs up to 25 volts @ 4 Amps. I was limited to building a 4 Amp power supply due to the transformers available. All power supplies built and tested were stable and had low ripple output. You can add a 1.0 uF tantalum capacitor across the output to ensure low output impedance at high frequencies. ■

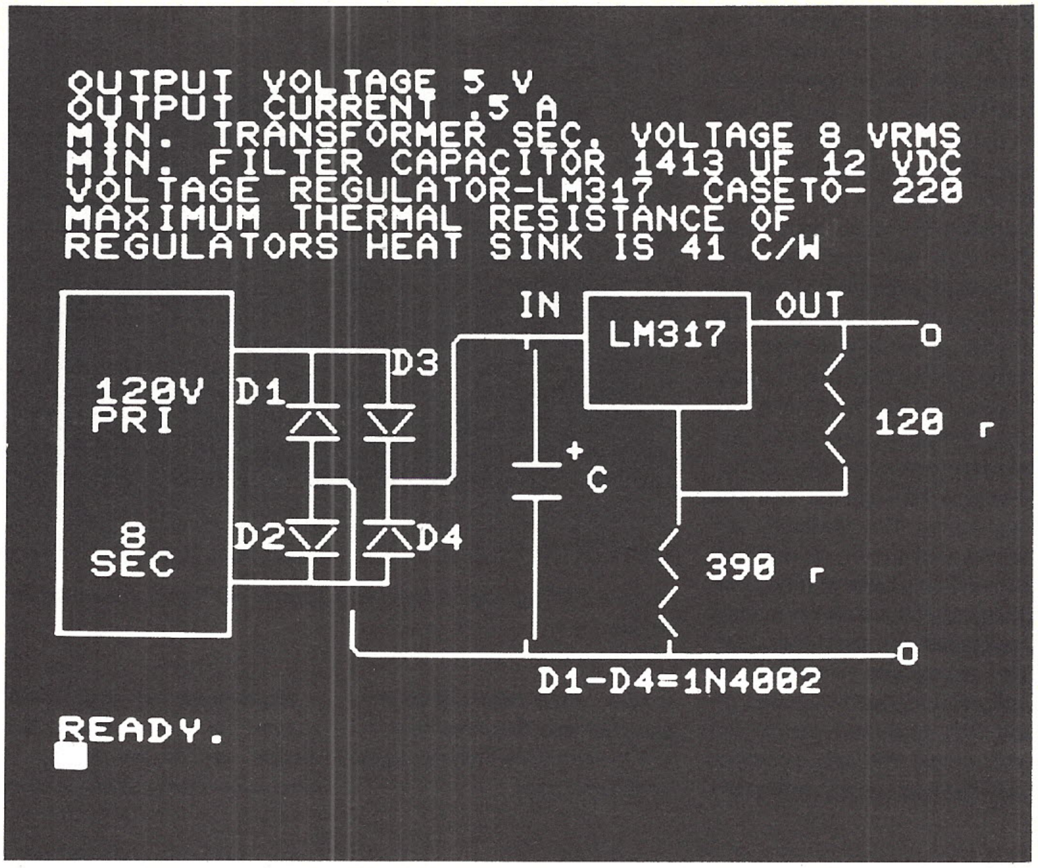


Photo 2. The author's PET showing a power-supply design using a Darlington connection for higher output current.

```
300   NEXT I
303   REM DETERMINE VALUE OF R1 AND R2.
304   FOR J=1 TO 2
305   RX=(VOUT/1.2-1)*R1
310   FOR I=1 TO 38
315   DELTAR (I) = ABS (RX-RES(I))
320   IF DELTAR (I)>DELTAR(I-1) THEN GO TO 330
325   RS(J)=RES(I)
330   NEXT I
335   R1=120
340   NEXT J
350   RATIO=VOUT/1.2-1
355   R210=RS(1)/100
360   R312=RS(2)/100
365   D1=ABS(RATIO-R210)
370   D2=ABS(RATIO-R312)
375   IF D1<D2 THEN GO TO 395
380   R1=120
385   R2=RS(2)
390   GO TO 405
395   R1=100
400   R2=RS(1)
405   IF IL>0.5 THEN GO TO 425
410   IF PCAL>5 THEN GO TO 430
415   THETA=INT(100/PCAL-6)
416   SERTR=0
418   TO=220
419   CASE $=T$
420   GO TO 465
425   IF IL>1.5 THEN GO TO 450
```


✓336

```
522  NEXT I
```

```
43Ø   IF PCAL>12 THEN GO TO 45Ø
435   THETA=INT(1ØØ/PCAL-3.4)
44Ø   CASE $=K$
441   TØ=3
445   GO TO 465
45Ø   SERTR=INT(Ø4+PCAL/20)
452   IF SERTR<1 THEN SERTR=1
455   THETA=INT(1ØØ/(PCAL/SERTR)-2.4)
46Ø   CASE $=T$
461   TØ=22Ø
465   PRINT"cs__________SPECIFICATIONS"
47Ø   PRINT"OUTPUT VOLTAGE "VOUT"V"
475   PRINT"OUTPUT CURRENT"IL"A"
48Ø   IF CT=1 THEN GO TO 495
485   PRINT"MIN. TRANSFORMER SEC. VOLTAGE"VRMS"VRMS"
49Ø   GO TO 5ØØ
495   PRINT"MIN. TRANSFORMER SEC. VOLTAGE IS"2*VRMS"VRMS CT"
5ØØ   PRINT"MIN. FILTER CAPACITOR"C"UF"CVD"VDC"
5Ø1   V=RMS*(CT+1)
5Ø5   PRINT"VOLTAGE REGULATOR-LM317 CASE"CASE$"TO-"TØ
51Ø   PRINT"MAXIMUM THERMAL RESISTANCE OF"
512   X=SERTR
513   IF SERTR>=1 THEN GO TO 517
515   PRINT"REGULATORS HEAT SINK IS"THETA"C/W"
516   GO TO 536
517   PRINT"TRANSISTOR'S HEAT SINK IS"THETA" C/W"
518   RK=INT(VOUT/.Ø3)
519   FOR I=1 TO 25
52Ø   DELTAR(I)=RES(25-I)-RK
521   IF DELTAR(I)<=ØTHEN GO TO 523
522   NEXT I
523   R3=RES(25-I)
53Ø   PRINT"NUMBER OF EXTERNAL 2N3Ø55 OR EQUIVALENT"
531   PRINT"TRANSISTORS REQUIRED IS "SERTR
532   PRINT"cd TYPE 1 AND RETURN TO SEE SCHEMATIC"
533   INPUT B:IF B=1 THEN GO SUB 4Ø
534   IF CT=1 GO TO 65
535   GO TO 61Ø
536   IF CT=1 THEN GO TO 539
537   GO SUB 4
538   GO TO 61Ø
539   GO SUB 22
54Ø   GO TO 61Ø
545   PRINT"cs DESIRED OUTPUT VOLTAGE EXCEEDS 25 VOLTS"
546   PRINT"cd TYPE IN YOUR DESIRED OUTPUT VOLTAGE"
555   GO TO 11Ø
56Ø   PRINT"cd DESIRED OUTPUT VOLTAGE IS LESS THAN"
561   PRINT"THE MINIMUM OF 1.2 VOLTS"
57Ø   GO TO 546
59Ø   PRINT"DESIRED LOAD CURRENT EXCEEDS 5 AMP"
591   PRINT"MAXIMUM"
595   GO TO 125
6ØØ   PRINT"cs SPECIFIED TRANSFORMER IS TOO SMALL FOR"
6Ø1   PRINT"THE DESIRED OUTPUT OF "VOUT" VOLTS"
6Ø4   PRINT"cd TYPE IN TOTAL RMS SECONDARY VOLTAGE".
6Ø5   GO TO 199
6Ø6   PRINT"cd LOAD CURRENT TOO SMALL TO UTILIZE THE__
      CAPABILITIES OF THE LM 317"
6Ø7   GO TO 135
61Ø   END
```

Get Your PET on the IEEE 488 Bus

Part 2 of this "opus computerus" examines the file characteristics of the IEEE 488 bus.

Gregory Yob
Box 354
Palo Alto, CA 94301

Your PET has a "built-in" way of communicating through the IEEE 488 bus. In BASIC, the IEEE 488 looks like a file—just as the cassettes are files. The OPEN statement is used to specify a physical device number of 4 to 30, and the open logical file now talks via the IEEE 488 bus.

A complete understanding of PET tape files is a prerequisite for working with the IEEE 488 as a BASIC file. An article in the January 1979 *Kilobaud Microcomputing* ("PET Techniques Explained") covers many "innocent" errors that will result in mysterious malfunctions.

IEEE 488 Information Transfers

Talking to a Device.

1. OPEN a BASIC file to the device's address. For example, OPEN 1,4 will open the IEEE bus to device 4. Your BASIC program will see this as file #1.

2. PRINT# to your OPENed file. PRINT#1,"HELLO, DEVICE" will address the device to listen, send the string HELLO, DEVICE, add a carriage return with EOI true and then issue the UNT (Untalk) command.

3. Repeat step 2 as needed. Note that after each PRINT#, the IEEE bus is free, since the UNT has been sent.

PRINT# will send the same characters, including the skip character after numbers, as PRINT does to the screen. If you want to send several items, be sure that any needed delimiters, such as ",", are included.

Listening to a Device.

1. OPEN a BASIC file to the device's address.

2. Use INPUT# or GET# to fetch a line or a character from the IEEE bus.

3. Check the status word, ST, for an error, such as time-out. If the device is slow, the PET will complete the INPUT# or GET# and put a nonzero value into ST, which must be checked immediately after the I/O operation. If ST indicates a time-out, jump back to step 2.

4. Convert the data from the INPUT# or GET# as needed, and if more is needed, go to step 2.

Note that after each INPUT# or GET#, the UNT command is sent to the IEEE bus. This will truncate long messages from the device, especially with GET#. Also note that INPUT# (string) and GET# (string) work the best. The BASIC string functions (MID$, RIGHT$, LEFT$ and VAL) will help you get the data into a usable form.

Talking to More than One Device.

1. OPEN a file for each device.

2. Using CMD, send a dummy message to each device. For example, CMD 1:CMD 2:CMD 3 will set up each device (as specified in the OPENs for files 1, 2 and 3) by sending carriage returns to the devices and leaving them as listeners on the bus.

3. PRINT# to the IEEE bus. Any of the OPENed files may be used.

4. Repeat steps 2 and 3 as needed. Since PRINT# ends with the UNT, step 2 must be repeated after each PRINT#.

Transfer from One Device to Another.

1. OPEN a file for each device.

2. CMD to the device that is to be the listener.

3. INPUT# from the device that is to be the talker.

4. Repeat step 3 as needed.

INPUT# does not send a UNL, so the device that was CMDed remains on the bus as a listener. All information sent by the talker to the PET is also received by the listener. To turn off the listener, use a PRINT# to the listener's file. If the talker is slow, check ST and repeat step 3 as required.

LISTing a BASIC Program to a Device

1. OPEN a file to the device.

2. CMD to the device.

3. Enter the LIST command.

4. When the LIST is finished, do a CLR.

The PET's graphics and cursor characters will not print correctly on a standard ASCII printer. (I have a BASIC listing program available.)

The best way to learn the PET files and IEEE 488 is by specific examples. After a detour through CMD, we will continue with two examples. These should provide you with enough information to get started. If you have no success, refer to the section on Common Errors (found later in this installment).

CMD

CMD is an unusual PET command. Consider its functions:

1. Anything that BASIC wants to say is now routed to the device that CMD's file number refers to. If this isn't the screen, nothing that BASIC says will appear on the screen.

2. If a list of variables and literals is provided after the CMD, they will be sent to the device in the same way as PRINT# will.

3. However, if the device is on the IEEE bus, no UNL will be sent, so the device will remain in the listening state and receive any following data sent on the IEEE bus.

To see how CMD operates, get two scratch tapes and enter the program in Example 1. Now SAVE and VERIFY this program on one of your tapes. Put the other tape in the tape unit and execute the following:

```
OPEN 1,1,1
PRESS PLAY & RECORD ON TAPE#1
```

Perform this and wait until the tape stops.

```
OK
READY.
```

```
10 REM CMD EXAMPLE
20 PRINT"****"
30 OPEN 1
40 GET#1, A$
50 PRINT A$;
60 IF A$=CHR$(90)THEN PRINT"****":END
70 GOTO 40
80 REM Z
```

Example 1.

Now enter CMD 1. Note that READY. didn't appear; it was provided by BASIC and is now residing in the tape buffer. The cursor is blinking below the C in CMD. Continue with:

```
LIST
CLOSE 1
CLR
READY.
```

Note that the CLOSE 1 didn't get the READY. back. It took the CLR to return BASIC's messages to the screen. If you enter LIST, the program will appear on the screen. Rewind the tape and RUN. Three asterisks now appear after the RUN. These were printed by the program. This is one reason I don't trust my PET after a CMD. The text between the OK and the ending READY was found as a data file.

When the PET was under the influence of CMD, the letters you typed in were put onto the screen. This echoing is done by the PET's operating system, so CMD won't put these out to the device.

Though CMD looks like a good way to LIST program to tapes as data files, there is a snag. My example is shorter than 191 characters, and a LIST via CMD isn't smart enough to "jiffy" the data tape (this has been fixed on the new PETs). You run the risk of losing tape records when you try to read an "unjiffied" tape.

Try to verify that CMD 1,"HELLO OUT THERE" will print HELLO OUT THERE onto the tape. Remember that if you CMD a device on the IEEE 488 bus, any PRINT# to the bus will require a repetition of the CMD if you want the device to remain in the listening state.

Talking to the Clock Again

(For a description of the HP clock see part 1 of this article.)

First, you must check the device address on the DIP switch (which will be near the 488 female connector) and make sure the address is in the range 4 to 15. The enter a short program (Example 2) into the PET. This program consists of three subroutines to facilitate communicating with the clock. Remember that the PET will not accept an INPUT statement as a direct command.

```
10  OPEN 1,7
20  RETURN
100 INPUT"SAY TO CLOCK:";S$
110 PRINT#1,S$
120 RETURN
200 INPUT#1,C$
210 PRINT"CLOCK SAYS: ";C$
220 RETURN
```

Example 2.

First, enter GOSUB 10 as a direct command. This opens file 1 to device 7, which is our clock on the IEEE bus. OPEN merely sets things up; nothing is sent to the bus yet.

To read the time, enter GOSUB 200:

```
GOSUB 200
CLOCK SAYS: 0103020204 (Jan. 3, 2:02:04 AM)
```

Your PET might give ? SYNTAX ERROR after this operation. This is a harmless feature of the PET.

To set the clock, using Jan. 29, 9:17 PM, as our example, enter:

```
GOSUB 100
SAY TO CLOCK? RDDDDDDDDDDDDDDD
DDDDDDDDDDDDD(28 Ds)
```

The clock starts at day 1. To set to day n, use n – 1 Ds. To set the hour, enter the following.

```
GOSUB 100
SAY TO CLOCK? HHHHHHHHHHHHHHH
HHHHHH(21Hs)
```

Minutes and seconds are set similarly.

```
GOSUB 100
SAY TO CLOCK? MMMMMMMMMMMMM
MMMMSSS (17Ms, 3Ss)
```

We are now set to 9:17:03. When I did this by hand, the clock moved forward about a minute, so the number of M's used should be changed to accommodate for this.

Talking to the HP 8165A Programmable Signal Source

(For a description of the HP 8165A, see part 1 of this article.)

The 8165A is a fine instrument with many switches, knobs, buttons and options and a correspondingly wide array of IEEE 488 commands (see Fig. 12, part 1).

The precise contents of each example concern the 8165A, which is an instrument you will probably never meet! My intention is to show you how direct

mode commands—that is, BASIC statements without line numbers—can be used to control an instrument and help in debugging.

First, I hooked the 8165 to the 488 cable, and the PET turned on. The 8165 was addressed to 8. When the PET came on, IFC was true for about one second. This put the 8165 in local mode, where the front panel works as usual. Many instruments will ignore their front panels when the 488 bus addresses them. Once the PET addresses the 8165, you cannot control it from the front panel anymore. (An LED indicates this on the 8165.)

The following short program takes care of input from the instrument:

```
10  INPUT#1, A$
20  PRINT A$
```

This substitutes for the illegal direct command (INPUT#1,A$: PRINTA$), which I would like to use, but the PET forbids (try it and see!).

Since I wanted the 8165 to output a 1 kHz sine wave at an amplitude of 1.5 volts, I used the following IEEE commands:

F1—Set to sine wave
FRQ 1 kHz—Set frequency
AMP 1.5 V—Set amplitude
I1—Set to normal operation (continuous signal output)

First, open the IEEE file:

```
OPEN 1,8
READY.
```

Then send the settings:

```
PRINT #1,"F1"  (At this point, the "Remote" LED went on, and I can no longer work the front panel.)
PRINT #1,"FRQ1KHZ"
PRINT #1,"AMP1.5V"
PRINT #1,"I1"
```

Nothing happened! My scope showed only a flat trace! Upon reviewing my steps, I noticed that I overlooked the Disable Output (OD) and Enable Output (OE) commands. I entered PRINT #1,"OE", and a sine wave appeared on the scope.

You could also send this setting as one string. For example, PRINT #1,"F2FRQ1.2KHZAMP 1.2VI1OE" sets up a 1.2 kHz triangle wave at 1.2 V amplitude.

The 8165 can also report some of its switch settings. Now we can use the tiny program in the PET:

```
GOTO 10
F1 D2 I2 FM0 AM0
```

Since the PET has difficulty with GOSUB in direct mode and the IEEE bus, we must make a program change:

```
10  INPUT#1, A$
20  PRINT A$
30  RETURN
```

We will quickly be reminded that any time we change a program, all the variables, including opened files, will be lost:

```
GOSUB 10
?FILE NOT OPEN ERROR IN 10
```

So we try again:

```
OPEN 1,8
GOSUB 10
F1 D2 I2 FM0 AM0
?SYNTAX ERROR IN 22066
```

The PET will provide the ?SYNTAX ERROR about 90 percent of the time when the IEEE is accessed via the INPUT# statement and the PET is executing a directly called subroutine. However, this doesn't appear to affect anything. I avoided this by not making the little program a subroutine the first time.

So, if you are in a pinch, remember that the PET's direct command capability can rescue you with IEEE 488 devices and provides an inexpensive way to explore a new instrument.

Talking to More than One Device

Now that each of the instruments has been in the bus individually, the next step is to try the 488 with both of them on at the same time. I connected the HP clock and the 8165 to the 488 bus and gave the clock address #7, and the 8165 address #8. Then I entered the short program for INPUTs:

```
10   INPUT #1, A$
20   PRINT A$
30   END
100  INPUT #2, B$
110  PRINT B$
120  END
```

First, OPEN the files:

```
OPEN 1,7
OPEN 2,8
```

If you get a ?FILE OPEN ERROR, just enter CLR and start over.

Taking a peek at the clock resulted in:

```
GOTO 10
0130051957     (30 Jan., 5:19:57)
```

And peeking at the 8165 gets me:

```
GOTO 100
F1 D2 I2 FM0 AM0
```

which is the usual mystery message that the 8165 says to me. There isn't any point in explaining this message, for your instrument will say something different and meaningful only to you.

PRINT #1 and PRINT #2 will work just fine, and so two instruments and the PET can live in harmony together.

A Gotcha

I decided to turn off the 8165 with the PET set up for two instruments as described above. Sure enough, strange things happened.

The clock worked fine:

```
GOTO 10
0130052525
```

And just for fun, look what happens with the 8165 (which isn't on):

```
GOTO 100
F1 D2 I2 FM0 AM0
```

The 8165 has some internal batteries to store and memorize settings until it is turned on again. It also will respond to the IEEE 488 bus.

Now to try things in reverse—the clock doesn't have any batteries. (Clock is off; 8165 is on.)

```
GOTO 100
F1 D2 I2 FM0 AM0     The 8165 is fine
GOTO 10
F1 D2 I2 FM0 AM0     What's this?
```

The 8165 will reply to any address if it is the only device on the bus. The clock acts in the same way. (I don't know if this is a PET fault or an HP design decision. Check your device.)

If your program is intended for more than one device, this can be a disaster. Make sure all required devices are operating when using multiple devices on the bus.

I ran into another gotcha: the 8165 wouldn't accept every frequency change. I tracked this problem down to the presence of the HP clock on the bus. When I turned the clock off, everything worked fine. When debugging, remember to have only one device on your bus.

Common Errors

In theory, if you have under-

MARK GORDON COMPUTERS

A DIVISION OF MARK GORDON ASSOCIATES, INC. ✓84

15 Kenwood Street, Cambridge, Massachusetts 02139, (617) 491-7505

SORT-80

Produced exclusively for Mark Gordon Computers by SBSG

TRS-80* disk files may be sorted and merged using SORT-80, the general purpose, machine language, sort program. Written in assembly language for the Z-80 microprocessor, it can:

- Sort files one disk in length
- Sort Direct Access, Sequential Access and Basic Sequential Access files
- Reblock and print records
- Recontrol files from disk
- Be executed from DOS
- Be executed from BASIC
- Be inserted in the job stream
- Allow parameter specification
  - input/output file specification
  - input/output record size
  - lower/upper record limit
  - print contents of output file
  - input/output file key specifiers

The minimum requirement is a 32K TRS-80* Level II computer with one disk drive or a single drive Model II computer. It will operate on 35, 40 and 77 track drives, and has been tested on TRSDOS 2.1, 2.2, 2.3, NEWDOS 2.1, 3.0, and VTOS 3.0.1. It is compatible with most machine language printer drivers. Sort time is fast; for example, a 32K file will sort in approximately 40 seconds. **$59**.

PCS

Program Catalog System from SBSG

This menu driven system provides the TRS-80* user with a computerized method to keep track of all programs and data files. The idea is to build and maintain on a file a disk detailing each program including program name, size, creation date, and a brief narrative as to function. Programs are provided to:

- create, update, or display
- print in disk number order
- print in alphabetical order
- print file listing
- create a file automatically

With a 32K system you can catalog 150 programs; with a 48K system you can catalog 300 programs; or you can catalog 650 programs without sort. **$29**

*TRS-80 is a registered trademark of Radio Shack, a division of Tandy Corp.

InfoBox

The information manager

InfoBox is the easiest-to-use information manager available for the TRS-80*. It's ideal for keeping track of notes to yourself, phone numbers, birthdays, inventories, bibliographies, computer programs, music tapes, and much more. This fast assembly language program lets you enter free-format data, variable length items and lets you look up items by specifiying a string of characters or words that you want to find. You can also edit and delete items. Items entered into InfoBox can be written to and read from cassette and disk files. All or selected items can be printed on a parallel or serial printer. InfoBox occupies 3K. Specify cassette or disk version. Special introductory price **$24.95** until June 15; **$29.95** after.

DBUG+

The ultimate monitor/disassembler

Compare the features and price of **DBUG+** with other monitor/disassembler programs. It offers nine true, single-byte breakpoints, single step program execution, hex and decimal arithmetic including multiply and divide and conversions, ASCII dump that distinguishes all 256 codes, disassembly to screen and printer in full Zilog mnemonics, and register set command. It also has the usual port I/O, hex and decimal memory dump, change, move, copy and exchange memory features offered by others. Ideal for the user who wants to experiement with assembly language or to write subroutines to call from BASIC; essential for the serious programmer. Special introductory price **$24.95** to June 15; **$29.95** after.

FMS

File Management System by SBSG

This menu driven program allows you to define and create files for your own use. You can;

- sort these files in;
  - ascending order
  - descending order
  - on up to 3 seperate fields
- scan the files
- summarize any numeric or dollar data fields
- print the field records
- create, add to or delete field records

$49.00

Model II versions of SBSG software available. Dealer inquiries invited.

stood everything to this point, you can now get an IEEE 488 instrument and make it play with your PET. In practice, this won't happen.

Finding errors is the hardest part of programming, and when you work with the IEEE bus, you can make many mistakes that don't look like errors. When you are able to see errors easily and immediately, you won't need this article.

Here is an incomplete list of the common errors in wait for the unwary IEEE/PET programmer.

The misplaced address. The PET's IEEE addresses are from 4 through 30. The addresses 0 to 3 are reserved for the PET's other I/O devices:

0—Keyboard
1—Tape unit #1
2—Tape unit #2
3—Video screen

If you OPEN a file to the reserved addresses, you won't be speaking to the IEEE bus!

If a device isn't running when the PET wants to talk to it, you will usually get a ?DEVICE NOT PRESENT ERROR. However, if some other device is operating on the bus, you might get the other device's response instead. This happened to me with the HP clock and the 8165. If one was turned off, the other would respond, even though the OPEN statement was referring to the inactive device. This can badly confuse your program.

Time-outs. The PET will only wait for 64 milliseconds before giving up on a device that is slow to respond to the IEEE 488 handshake. Though the IEEE 488 is supposed to work at *any* speed, you may wonder what to do if a device on the bus has failed. If the PET were to wait for a response, there would be no way to return to the user. The 64 ms interval was chosen from the timers available on the 6522 VIA chip, which can count up to 65535 at the 1 MHz clock rate of the PET.

Most instruments will respond within the 64 ms interval, and the PET will read and write the data correctly. This was true of the HP instruments at my disposal. To exercise the PET time-outs, I attached both the clock and the 8165 to the bus, and then OPENed a file to a nonexistent address:

```
NEW

10  INPUT#3,A$
20  IF ST THEN PRINT"ST IS" ST
30  PRINT A$
40  A$=""

OPEN 1,7     (Open the clock to file 1)

OPEN 2,8     (Open the 8165 to file 2)

OPEN 3,10    (The nonexistent device)
```

The little program attempts to input from the nonexisting device. The ST value is a reserved BASIC variable used by the PET for indicating I/O conditions. If ST isn't zero, something went awry.

Now to talk a bit to the devices to wake them up:

```
PRINT #1,"R"    (And the clock resets)
PRINT #2,"E0"   (And the 8165 puts out
                 a signal)
```

If a look at ST is made, all's well:

```
PRINT ST
 0
```

This may take a few tries to work right.

Now to try that nonexistent device:

```
PRINT #3, "HELLO"
```

Looks OK, right? Well, let's see . . .

```
PRINT ST
-128
```

This is the PET's ST code for "device not present."

Now to try the little program:

```
GOTO 10
ST IS 2

READY.
```

The ST code is 2, which is the time-out for reading data; the nonexistent device didn't say anything. Recall that line 30 said to print A$. The PET *did* print A$, which was an empty string.

The solution to this dilemma is to keep on trying! Write a loop that redoes the INPUT# or PRINT#. In most cases, a slow device will send its characters rapidly enough—once it has its message ready.

Consider these two sample loops:

```
100  PRINT #5," some message or other"
110  IF ST=1 THEN 100
200  INPUT #6,B$
210  IF ST=2 THEN 200
```

If you want to mask for certain bits, you can use the AND operator, but parentheses are needed. The above examples would read:

```
110  IF (ST) AND 1 THEN 100      and
210  IF (ST) AND 2 THEN 200
```

The removal of the parentheses makes the PET see the expression as:

```
IF ST AND 1    looks like    IF S TAN D 1
```

which will result in a ?SYNTAX ERROR. Use parentheses or rearrange the order of operations in these cases.

The literal principle. PET outputs to a file the same characters that it sends to the screen. This is also true for the IEEE 488. The PET's format for PRINTing a number is:

(space or − sign) (digits) (optional exponent) (cursor right)

The PET IEEE 488 File I/O Statements

The PET sees the IEEE 488 bus as a file, and the file I/O statements apply to IEEE 488 transfers. Be sure you know the cassette file I/O before tackling the IEEE 488 bus.

The PET file I/O statements are:

● OPEN (file number), (device number), (secondary address), (filename)

OPEN instructs the PET to associate the file number with the desired I/O device. BASIC uses the file number in its PRINT#, INPUT# and GET# statements to determine where the I/O is to take place. The file number may be from 1 to 255.

The device numbers are assigned as follows:

0—Keyboard
1—Cassette unit #1
2—Cassette unit #2
3—Screen
4—30 IEEE 488 bus

This implies that your IEEE device must be addressed in the range of 4 to 30. Most IEEE devices have a switch or jumpers that permit the changing of their addresses.

The secondary address and filename are optional. However, if you want to use the filename, the secondary address must also be included. The secondary address has the range of 0 to 31.

If the filename is not specified, the OPEN statement sends nothing to the IEEE 488 bus. When BASIC sees the PRINT#, INPUT# and GET# statements, the device number (and secondary address, if specified) are put on the IEEE bus as part of the usual transfer sequences.

If a filename is specified, (i.e., A$ or "SOME NAME"), the OPEN statement activates the IEEE bus making ATN true and sends:

LISTEN (to the appropriate device)
SECONDARY ADDRESS (ORed with 11110000)
FILENAME (all characters)

This permits suitably complex command sequences that require ATN to be true to be sent. If the command sequence has to be repeated later, CLOSE the file and OPEN it again. I haven't been able to check if the above assertions about the filename are true. If you have a bus analyzer, check this out!

● PRINT# (file number), (values to be sent)

First, don't use the abbreviation ?# ; it won't work (when executed, you will see ?SYNTAX ERROR) and will list as PRINT#. Spell out PRINT completely!

The PRINT# sets ATN true and sends the device number as a LISTEN address. If a secondary address as specified, it will be sent also. The device number and secondary address are taken from the appropriate OPEN statement.

ATN is then made false, and the values to be sent are transmitted as ASCII characters in exactly the same way as they would be sent to the screen. For example, if a number is sent, a cursor right character follows the last digit. If you use "," to separate columns, lots of cursor rights are sent. If the PET feels a number should be in scientific format (i.e., 1.53E−07), that's what is sent! EOI is made true with the last character of data sent.

After the values are sent, an UNLISTEN is sent (with ATN true), and all listening devices are set free.

● INPUT# (file number), (values to be input)

INPUT# sets ATN true and sends the device number as a TALK address. If a secondary address was specified, it will be sent too. The pertinent OPEN statement is used for these values.

ATN is then made false, and the PET accepts characters from the device to the PET's input buffer. If the talker activates EOI, a carriage return is added to the end of the buffer.

After the characters are accepted and carriage return or EOI is recognized, the PET sets ATN true and sends an UNTALK, which releases the device.

BASIC then scans the input buffer in the same way that an ordinary INPUT statement looks at what is typed in. This means that commas and quotes will have the same effects as with normal INPUT. It is best to use an INPUT (string) form and hope your device doesn't send any commas!

As with cassette INPUT#, an 80-character buffer is used. If more than 79 characters arrive without a carriage return, the PET will go into "limbo," and all is lost. (New PETs have this fixed. Over 80 characters are ignored (or worse, the buffer is initialized, and the first 80 characters are lost!). If you have a new PET, try it with cassettes and find out what happens.

INPUT# is susceptible to "time out," and ST should be checked for a time out. Repeat the INPUT# if a time out is detected.

● GET# (file number), (value for entry)

GET# sets ATN true and sends the device number as a TALK address and the secondary address, if specified. ATN is made false, and a single character is accepted.

Then, the UNTALK with ATN true is sent, and the character given to BASIC. For the reasons that make GET X unusable, be sure to only use the GET# (string) form.

The assertion of the UNTALK after GET# makes transmission of multicharacter messages from devices impractical, as most devices will try to repeat their message on repeated application of GET#.

As with INPUT# ST should be checked for a time out, and if timed out, the GET# should be repeated.

● CLOSE (file number)

CLOSE releases the I/O assignments. The PET will allow a maximum of ten files OPEN at one time, and CLOSE will let you reuse an I/O assignment. If you OPEN more than ten files, old PETs will go into limbo and all will be lost. New PETs presumably have this fixed.

If the corresponding OPEN statement had a filename specified, CLOSE sets ATN true and sends the device number and secondary address (ORed 11100000). This feature is intended for PET peripherals.

● CMD (file number), (values to be sent)

CMD initiates the same sequence as PRINT# and sends the values, if any, in the same way that PRINT# does. When finished, CMD does not send the UNLISTEN, so any devices addresses with CMD will listen to further CMDs or PRINT# to the IEEE bus.

All of BASIC's output will be routed to the device defined in the OPEN statement for the file number. If the PET is in command mode, this includes the READY., error messages and LIST. If in run mode, any BASIC printouts, from PRINT to the screen, will go to the IEEE bus instead. A PRINT# will recover from the effects of CMD.

If you are using CMD in command mode, the cursor may not echo the RETURNs you press. The PET will "echo" your keystrokes, but any outputs from BASIC will vanish to the IEEE device. The PRINT# to your IEEE device is the safest recovery from CMD. Remember that any editing of a BASIC program will destroy all variables. This includes open files and CMDs.

● ST (status word)

After each I/O operation, the PET sets the value of a special variable named ST, which will hold its value until the next I/O operation. So the best policy is to check it immediately! The values of ST for the IEEE bus are:

1 Timeout on write
2 Timeout on read (This one should *always* be checked)
64 EOI true
−128 Device not present

The PET waits for 64 milliseconds to see if a device will respond to the IEEE handshake. If the device doesn't, the I/O operation is quietly aborted, and ST is set. If you are INPUT#ing, you will get "nothing" or zeroes back. If you are PRINT#ing, everything seems to be all right. If your device is slow to respond, checking ST is mandatory.

PRINT#, INPUT# and GET# will return the ?DEVICE NOT PRESENT error if the bus is in an illegal state (which is true if the bus has no devices or the LISTEN or TALK isn't responded to). ST will also be set.

● LOAD, SAVE and VERIFY

The old PETs have a severe error in their IEEE software which prevents the functioning of LOAD, SAVE or VERIFY. The ATN line was left true during the data part of the transfer. This is why owners of old PETs who purchase the PET disk get the new ROMS; the disk won't function with the old ROMs.

The format is the same as with tapes:

LOAD (filename), (device number)
SAVE " " "
VERIFY " " "

Once the IEEE bus is set to listen or talk, the first four bytes must contain the beginning and ending address + 1 of the block to be transferred. The transfer is then done as pure binary until finished. The bus is then released with an UNT or UNL as needed.

VERIFY will say ?VERIFY ERROR and set ST to 16 if any mismatches were found between the incoming data and the core image in the PET's memory. Since my PET is an old model with the original ROMs, I haven't been able to check LOAD, SAVE and VERIFY for the IEEE 488 bus.

This can raise havoc with an IEEE device that is expecting a character after the number.

Consider the following example:

```
10  PRINT "clr";    (clear screen)
20  FOR J = 1 TO 10
30  PRINT "*********************"
40  NEXT J
50  PRINT "hm";    (home cursor)
60  FOR J = 1 TO 10
70  PRINT J"IS A NUMBER"
80  NEXT J

RUN

 1*IS A NUMBER*****
 2*IS A NUMBER*****
 3*IS A NUMBER*****

 etc
```

The asterisk after the number comes from the cursor right character that was sent to the screen. The cursor right follows

any numbers sent to the IEEE 488 bus.

The following program sets the frequency of the 8165.

```
10  OPEN 1,8   (The 8165 is at address 8)
20  FOR J = 1000 TO 2000 STEP 10
30  PRINT #1,"FRQ"J"HZ"
40  FOR K = 1 TO 1000
50  NEXT K     (This is a 3 second delay
               loop)
60  NEXT J
```

When this is RUN, the 8165 gives all signs of distress. The frequency appears on the front panel, but the LED that indicates correct entry stays blinking (not completed). Also, the scope shows no change. The PET screen blinks at intervals, indicating that EOI is made true now and then. (I suspect the instrument is making this happen.)

The following modification will fix this:

```
30  PRINT#1,"FRQ"STR$(J)"HZ"
```

The STR$ function converts a number to the string that would be PRINTed, without the cursor right at the end! The general fix for numbers is simple: convert *all* numbers to strings before putting on the IEEE 488 bus.

Fractions. Now that the frequency example is working right, how about trying some other STEP sizes. Here is a simple change:

```
20  FOR J = 1 TO 2 STEP .01
30  PRINT #1,"FRQ"STR$(J)"KHZ"
```

The J loop was changed to do the same thing, but in kilohertz. Line 30 was changed to reflect this. When RUN, it all works fine until about 1.25 kHz—the 8165 now shows 1.259 kHz instead of 1.260. A look at J gives us the clue we need:

```
BREAK IN 40      (Press STOP key)
PRINT J
 1.25999999
```

The PET slips up when computing with fractions . . . and this eventually shows up. The fraction .01 becomes a repeating binary decimal, and after repeated addition, the round-off appears as a slight reduction of the number being added to. In this case, 1.260 turns into 1.25999999.

Catching this is easy . . . if J were put onto the screen first!

```
35  PRINT STR$(J)
```

If you do this, the first "blow up" comes at 1.22999999. Now you are faced with a programming problem: how to get around nasty numbers. One way is to take the INT function, such as:

```
STR$(INT(J*100+.5)/100)
```

which rounds the number in the hundredths place. More complex tricks will be needed if the PET insists on scientific notation, such as

```
2.35E-03
```

PRINT your IEEE output onto the screen while debugging.

Next month, we will wrap up our three-part series with a further look at the programming style with the IEEE 488. ■

ATARI® PERSONAL COMPUTER SYSTEMS

A Warner Communications Company

ATARI® 800™

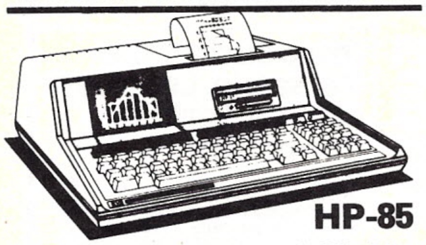
List $1080

ONLY $849

ATARI® 400™, List $630

OUR PRICE ONLY $499

820 PRINTER, List $599.95 .... **$499**
810 DISK DRIVE, List $699.95 .... **$589**



HP-85

Call for Price

- Extended BASIC Language
- Advance Graphics
- CRT Built-In Display
- Magnetic Tape Cartridge for Storage

CALCULATORS BY

HEWLETT hp PACKARD

HP-41C Calculator, "A System"	**$289.95**
HP-32E Scientific w/Statistics	**$ 53.95**
HP-33C Scientific Programmable	**99.95**
HP-34C Advanced Scientific Programmable	**123.95**
HP-37E Business Calculator	**58.95**
HP-67 Handheld Fully Advanced Programmable Scientific for Business & Engineering	**298.95**
HP-97 Desktop w/Built-in Printer	**579.95**

APPLE II, 16K, List $1195	**$ 989**
32K, List $1395	**$1169**
48K	**1259**
COMMODORE PET	**Call for Prices**

Prices do not include shipping by UPS. All prices and offers are subject to change without notice.

Personal Computer Systems

✓303

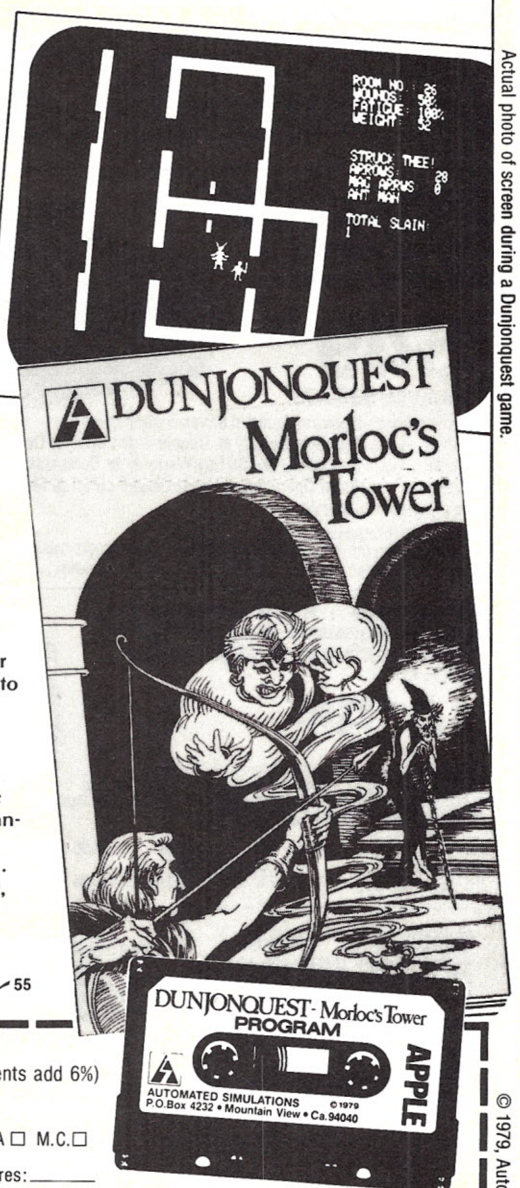
609 Butternut Street
Syracuse, N.Y. 13208
(315) 478-6800

YOU'RE FACE-TO-FACE WITH EVIL

It's Morloc The Wizard, the evil master of mayhem and illusion. He's threatening the village of Hagedorn.
Your mission is to kill Morloc and save the village in this dangerous REALTIME computer game from Automated Simulations.
Morloc lives in a 30-room Tower, where his minions and monsters do his bidding to create terrible hazards for you. You'll be attacked by Ogres, The Creeping Crud, Fire Elemental, Vampire Bats, Salamanders and his personal Genie. The fiend will even throw his devastating Fireballs at you.
And, just when you think you have him in your grasp, he'll teleport away.

He's fearsome and lethal.
If you can find the magical treasures in the Tower, you might be able to turn them against Morloc.
Try to find them. Then, if you do, try to decipher their meaning and use them.
Even worse, you're in REALTIME!

If you have a 24K PET, 16K TRS-80, or 48K APPLE, you can play "MORLOC'S TOWER" and come face-to-face with absolute evil.



Actual photo of screen during a Dunjonquest game.

Fantasy at Your Fingertips T.M.

HERE'S HOW YOU CAN TOUCH YOUR FANTASIES: Ask your dealer or rush $19.95 in check or money order to **Automated Simulations, Dept. TW1 P.O. Box 4232, Mountain View, CA 94040.**
Or, call our FANTASY LINE, toll free, **800-824-7888**, Operator 861 to place your order and to tell us what other fantasies you would like to touch. (California, call 800-852-7777, Operator 861. Alaska and Hawaii, call 800-824-7919, Operator 861.)

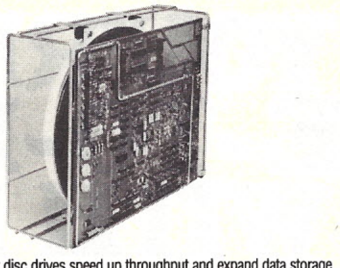
AUTOMATED SIMULATIONS

✓55

I want to **TOUCH** my fantasies...Rush me "MORLOC'S TOWER" for $19.95 (CA. residents add 6%)
I have: PET ☐ TRS-80 ☐ APPLE ☐
Payment enclosed ________ Bill my VISA ☐ M.C.☐
Account # ________ Expires: ________
Name ________
Address ________
City ________ State ____ Zip ________

GUARANTEE
If I'm not completely satisfied, I will send "MORLOC'S TOWER" back to you in 10 days for a full refund.

PRIAM Hard Disks Now Available from SIRIUS SYSTEMS!



PRIAM's high-performance, low-cost Winchester disc drives speed up throughput and expand data storage from 20 megabytes to 154 megabytes. And a single controller can be used to operate 14-inch-disc drives with capacities of 33, 66, or 154 megabytes or floppy-disc-size drives holding 20 and 34 megabytes. So it's easy to move up in capacity, or reduce package size, without changing important system elements or performance.

- *Fast, Linear Voice Coil Positioning*
- *10 ms track-to-track positioning*
- *Fully servoed head positioning*
- *Dedicated servo tracks*
- *DC Power required only!*
- *Simple, parallel Interface*
- *Optional SMD Interface*
- *50 ms Average Positioning time*
- *90 ms Maximum Positioning Time*
- *6.4 ms Average Latency*

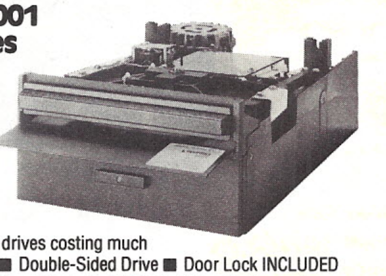
THE PRIAM LINEUP

Model/Disc Size	Capacity	Size	Weight	Price
DISKOS 3350 (14")	33Mbytes	7" × 17" × 20"	33 lbs.	**$2995**
DISKOS 6650 (14")	66 Mbytes	7" × 17" × 20"	33 lbs.	**$3749**
DISKOS 15450 (14")	154 Mbytes	7" × 17" × 20"	33 lbs.	**$4695**
DISKOS 2050 (8")	20 Mbytes	4.62" × 8.55" × 14.25"	20 lbs.	**$2995**
DISKOS 3450 (8")	34 Mbytes	4.62" × 8.55" × 14.25"	20 lbs.	**$3745**
DISKOS 570	5.3 Mbytes	floppy-size	(low)	**(low)**
DISKOS 1070	10.6 Mbytes	floppy-size	(low)	**(low)**

All PRIAM DISKOS Drives have a Transfer Rate of 1.03 Mbytes/Sec.
Optional SMD interface available for $150.

SIRIUS SYTEMS offer cases and enclosures for all PRIAM Hard Disk Drives. All 14" Winchester Drives will mount in our 14" Standard Case. The 8" Winchesters have two alternatives: a single drive case and a dual drive case. All SIRIUS SYSTEMS Winchester drive cases include Power Supply, internal cabling, switches, fan, extra AC outlet (not switched, but fused) and possess very adequate ventilation. Drive addressing is done on the rear of the Case and not on the drive iteself to provide ease of use during operation. All WINCHESTER DRIVE Cases are Warranted for a full year and come in our standard blue-black color scheme. Consult us for current availability and pricing.

Remex RFD 4000/4001 8" Floppy Disc Drives Double sided . . . Double density!!



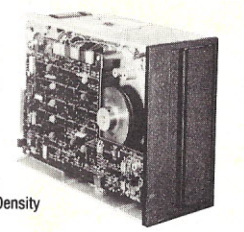
$549.95

RFD 4001, $569.95

Offers quality and features found in drives costing much more! ■ Single or Double Density ■ Double-Sided Drive ■ Door Lock INCLUDED ■ Write-Protect INCLUDED ■ 180 Day Warranty ■ Compatible with Shugart 850/851 ■ Low Power Operation ensures LONGER LIFE!! ■ Model RFD 4001 offers Data and Sector Separator

RFD 4000/4001 Technical Manual **6.95**
Connector Set #3 (AC, DC, Card Edge) . . **10.95**
Connector Set #4 (AC and DC) **2.95**
RFD 4000C/B Cabinet (for use with Power Modules) **29.95**

Remex 1000B . . . If you've been looking for a less expensive floppy disc drive, but not wanting to sacrifice quality—this is it!



$419.95

You get both in the Remex 1000B! For only $419.95 look at what you get: ■ 8" Floppy Drive ■ Single or Double Density ■ Hard or Soft Sectoring ■ Media Protection Feature ■ Single Density Data Separator ■ 180 Day Factory Warranty

Door Lock Option **$19.95**
Interface Adapter (REMEX-to-Shugart) . . . **$14.95**
Write Protect Option **$19.95**
Connector Set #1 (AC, DC, & Card Edge) . . . **$10.95**
RFD 1000B Technical Manual . **$5.95**
RFD 1000B CASE (for use with Power Modules) **$29.95**

SIRIUS 8" DISK POWER MODULES

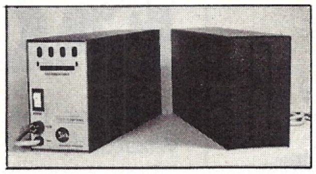
The Single and Dual Drive Power Modules are designed to provide DC and (switched) AC power for one (the Single Drive Power Module) or two (the Dual Drive Power Module—the DDPM will power three RFD 4000s or 4001s) 8" Floppy Disk Drives. Many features are included for safe and reliable operation and the Power Modules come with our standard 180 day WARRANTY (the Open Frame Power Supply warranty is for 2 years). All Power Modules will work with either the RFD 4000C/B or RFD 1000B case (color schemes match also).

Dual Drive Power Module (DDPM) **$139.95**

Single Drive Power Module (SDPM) **119.95**

SIRIUS 80+

Perfect Add-Ons for Your Computer System!



The SIRIUS SYSTEMS 80+ Series of Floppy Disk add-ons are designed to provide unmatched versatility and performance for your computer. Consisting of four different add-ons, there is a 80+ Series Floppy Disk to meet your need. All 80+ Series Floppy Disk are compatible with the TRS-80* and come ready to plug in!

COMMON CHARACTERISTICS

- *5 ms track-to-track access time*
- *Auto-eject*
- *180 day WARRANTY*
- *Exceptional speed stability — 1½%*
- *Single density (FM) or double density (MFM/M^2FM)*
- *Ultra high reliability*
- *2 year Power Supply Warranty*
- *Mix any or all 80+ Series on the same cable!*
- *Includes user accessible plugboard for drive reconfiguring*

SPECIFIC CHARACTERISTICS

The **SIRIUS 80+1** is a single sided, 40 track, highly reliable Floppy Disk add-on. Offering 5 more tracks than the Radio Shack model, it cost $140 less! Formatted data storage is 102K/20K bytes single/double density.

SIRIUS 80+1 . . . . **$359.95**

The **SIRIUS 80+2** is a dual sided, 70 track (35 per side), highly versatile Floppy Disk unit. It appears to the TRS-80* as TWO 35 track drives, yet COST LESS THAN HALF THE PRICE! Even greater savings result, since data is recorded on both sides of the media instead of only a single side. Using the plug board, it may be reconfigured for other computer systems! (The 80+2 operates as Drive 0 and any of the other three addresses (with the sandard Radio Shack Cable) or as any of four drives (with the SS Standad Cable).) Formatted data storage is 80.6K/161.2K bytes single/double density.

SIRIUS 80+2 . . . . **$449.95**

The **SIRIUS 80+3** is a single sided, 80 track, "Quad" density Floppy Disk unit. Offering 2⅓ times the storage of a Standard Radio Shack drive, the 80+3 greatly reduces the need for diskettes correspondingly. Additionally, because of the increased storage and faster track-to-track access time, the 80+3 allows tremendously increased throughput for disk based programs!!! The 80+3 INCLUDES SIRIUS's TRAKS-PATCH on Diskette. Formatted data storage is 204K/40K8 bytes single/double density.

SIRIUS 80+3 . . . . **$489.95**

The **SIRIUS 80+4** Floppy Disk add-on is a double sided, 160 track (80 per side), 5¼" monster! The ultimate in state-of-the-art 5¼" Floppy Disk technology, to 80+4 is seen by the TRS-80* as two single sided disk drives, each with 80 tracks. Thus, in terms of capacity one 80+4 is equivalent to 4⅔ standard Radio Shack drives — a savings of over 73% (not to mention diskettes!!!). (With a double density computer, the available memory is huge!) The 80+4 is similar to the 80+2 in that it arrives configured as Drive 0 and any of the other three addresses (with the standard Radio Shack Cable) or as any of four drives (with the SS Standard Cable). The 80+4 INCLUDES TRAKS-PATCH on Diskette. (The plug board is also included.) Formatted data storage is 408K single density or 816K bytes double density.

SIRIUS 80+4 . . . . **$624.95**

0

All 80+ Series Floppy Disk add-ons operate a 5 milliseconds track-to-track access time (eight times faster than the SA 400) but are Expansion Interface Limited to 12 milli-seconds for the TRS-80*.

*TRS-80© Tandy Corp.

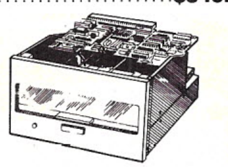
MPI 51/52 . . .

A Great Reliable Mini-Drive!

- *Fast! 5ms track to track access*
- *Exclusive Pulley-Band Design*
- *Unique Door/Ejector Mechanism*
- *Reliable 1½% Speed Stability*
- *Single/Double Density Operation*
- *Industry/ANSI Standard Interface*

MPI 51
(Single Head, 40 tracks, 120K/240K bytes Single/Double Density**) **$259.95**

MPI 52
(Dual Head, 70 tracks, (35/side), 218.8K/437.5K Single/Double Density**) **$349.95**



MPI 91/92 . . . NEW STATE-OF-THE-ART DISK DRIVE!

MPI 91
(Single Head, 80 tracks, 240K/480K Single/Double Density**) **$389.95**

MPI 92
(Single Head, 160 tracks (80/side), 480K/960K Single/Double Density**) **$499.95**

***Unformatted data storage*

Introducing the Versatile, Low-Cost OMEGA Series Controller

As new technological advances bring down the cost of fast, reliable mass data storage, the need for an inexpensive, versatile controller have become greater and greater. To meet this need, SIRIUS SYSTEMS' OMEGA Series Controller was designed.

The SIRIUS OMEGA Series Controller Module utilizes an **on-board microprocessor** to mediate data transfer to a wide variety of peripherals from an equally wide variety of host computer systems. Up to four Winchester Hard Disks (8" or 14"), four 5¼" Floppy Disks Drives and/or up to eight 8" Floppy Disk Drives may be **in use at one time.** Host systems interfacing is accomplished via a parallel or a serial interface. With the additon of a Personality module, the OMEGA Series Controller Module is directly **compatible with many popular computer systems** (among them the TRS-80*, Apple, Heath, and others). Provision is made for the addition of a streaming tape drive, also.

SPECIFIC HARDWARE FEATURES INCLUDE:

- *Control of up to twelve Floppy Disk Drives (eight 8" and/or four 5¼")*
  - *8" and/or 5¼" Disk Drive Utilization*
  - *Single (FM) or Double (MFM) density data storage*
  - *Hard or Soft sectored diskette usage*
  - *Utilization of "Quad" density (96 tpi) 8" or 5¼" Disk Drives*
- *Control of up to four WINCHESTER type PRIAM DISKOS Disk Drives*
  - *8" or 14" may intermix on the same cable*
  - *Accommodates 8" and/or 14" drives of 5.3Mbytes to 154Mbytes*
  - *Ultra-Fast data transfers*
- *Extremely flexible host-controller interfacing*

SPECIFIC SOFTWARE FEATURES INCLUDE:

- Dynamic format modifications via command words
- Extremely flexible format acceptance for unusual data storage formats
- Easily interfaces to standard operating systems (TRS-DOS*, CP/M***, etc)
- Operates in either get/put sector mode or data string mode
- Performance parameters may be changed by EPROM replacement or Dynaminic Reprogramming

Dedicated systems cards are also available on a limited basis for the STD-BUS and the S 100. These cards feature shared memory also (again, software selectable) in addition to the regular OMEGA Series Controller Module features. Consult SIRIUS SYTEMS for current price and availability for the entire line of OMEGA Series Memory Units and Controllers. **Dealer inquiries are invited.**

TO ORDER CALL (615) 693-6583

Phone Orders Accepted 9AM-7PM (ESDT)

We accept **MC, VISA, AE, COD** (requires Certified Check, Cashier's Check or Cash) and **Checks** (personal checks require 14 days to clear). **SHIPPING AND HANDLING:** $7.00 per Floppy Disk Drive or 80+ Module ■ 5% for other items (any excess will be refunded) ■ **Foreign Orders** add 10% for Shipping & Handling. Payment in U.S. currency ■ Tennessee residents add 6% Sales Tax ■ **VOLUME DISCOUNTS AVAILABLE**

SIRIUS SYSTEMS
7528 Oak Ridge Highway
Knoxville, Tennessee 37921 ✓67

Some Tips on Program Conversions

Lessons learned from converting a personal bookkeeping system.

Linda E. Bjelland
2288 Hollygrove Rd.
Memphis, TN 38138

A lot of software is available, but often it has to be converted from one system to another. I recently completed my first programming project on my new TRS-80 Level II 32K disk system, and it happened to be a conversion. It was an excellent tool for learning about my system and its capabilities. The program I converted was "Keepbook," a small personal-bookkeeping system written in North Star disk BASIC by Robert L. Marx (June 1979, issue, p.60).

My husband and I hoped to use our TRS-80 for household bookkeeping, but the financial systems we looked at were too complicated. "Keepbook" met our requirements and provided a good vehicle for me to learn something about using TRSDOS to handle disk files. The conversion took longer than I expected, but in the end I had a useful program.

As I completed the program and reviewed what I had done, I began to formulate some thoughts about conversion in general. Here are some of my rules; I'm sure you'll think of your own to add, based on your own experience with program conversions.

Understand the program before you try to convert it. It is surprising how often this rule is ignored. You should make sure that you understand specifically what the program is supposed to do. If the program came from a magazine or newsletter, there is probably an article or letter that explains its functions. Read it carefully, look at any examples that are provided and get a good functional feel for the program. If the program came from a "friend of a friend who has this terrific system . . . " get a sample run if possible, and get the name of the original author. If these aids aren't available (and sometimes even if they are), you are reduced to examining the code to discover the program's true functions. These are the times you pray for excellent documentation in remarks.

You should also know exactly what the program *doesn't* do. You may save yourself considerable time and effort converting something you won't use, or you may decide that it does some of what you want, but you prefer to write your own program.

Make a list of syntax substitution rules. You will probably do this, at least mentally, while you're reviewing the code. As long as you're thinking about it anyway, write it down. Many of these will be simple one-for-one substitutions. For example, in the published version of "Keepbook," a ¢ sign separated statements on the same program line; I had to replace this with a colon every time it occurred in my TRS-80 system. Some of the replacement rules may be more complicated to figure out.

In "Keepbook" I encountered the statement B9$ = CHR$(26) and, later, in several locations, PRINT#DO,B9$. From the author's remarks and the context, I soon figured out that the function was to clear the screen, for which I could use the CLS instruction. More complicated syntax and system differences may require rewriting sections of the code. I had to totally rewrite "Keepbook" disk I/O for the TRS-80, since TRSDOS handles random disk files with a different set of commands and formats than the North Star system.

Key in and run the program without any functional changes. Make only those syntax and technical changes you have on your list as you key in the program. Resist urges to make "small improvements"—adding a clarifying field to a display here, changing a variable name there. It is important to verify that you have a piece of code that performs the functions that you expect it to. If you fool with the code before it's tested, you may never know if the original version worked or not.

Carefully check and verify the output of this test run. Do not assume that because you received a display in the ex-

Keepbook program.

```
100 'KEEPBOOK BOOKKEEPING SYSTEM VERSION 1.0 JUNE, 1979
200 'ADAPTED FOR THE TRS-80 BY   LINDA E. BJELLAND
300 'FROM A NORTH STAR SYSTEM BY ROBERT L. MARX
400 'PUBLISHED IN KILOBAUD MICROCOMPUTING, JUNE, 1979
500 ' INITIALIZATION
600 CLEAR 250
700 CLS
800 DIM T$(35),F$(8),O$(1),T2$(50),L9$(30),L8$(62)
900 PRINT:PRINT:PRINT TAB(15);"'KEEPBOOK' BOOKKEEPING S
     YSTEM":PRINT
1000 PRINT TAB(24);"OPTIONS ARE:":PRINT
1100 PRINT TAB(5);"1=CONSTRUCT NEW FILE","2=START NEW Y
     EAR"
1200 PRINT TAB(5);"3=START NEW MONTH","4=ADD EXPENSE EN
     TRIES"
1300 PRINT TAB(5);"5=PRINT MONTHLY SUMMARY","6=H E L P"
1400 PRINT TAB(5);"7=PRINT SOURCE SHEET","8=QUIT"
1500 INPUT "OPTION = ";O
1600 IF O = INT(O) THEN 1700 :PRINT "ERROR - MUST BE IN
     TEGER":GOTO 900
1700 IF O>0 AND O<9 THEN 1800 :PRINT "ERROR":GOTO 900
1800 ON O GOTO 2000 ,4600 ,6000 ,7200 ,9400 ,12400,1430
     0,14000
1900 REM ******************************
2000 REM OPTION = CONSTRUCT FILE
2100 CLS: PRINT:PRINT:PRINT TAB(5);"CONSTRUCT NEW FILE"
2200 INPUT "HOW MANY ACCOUNTS ARE THERE ";A1
2300 IF A1<=0 THEN PRINT "MINIMUM 1":GOTO 2200
2400 MD$="N":GOSUB 15800
2500   LSET TI$(0)=MKI$(A1):LSET AC$(0)=MKI$(1)
2600   LSET S1$(0)=MKI$(1)
```

```
2700 FOR I%=1 TO A1
2800 GOSUB 18500
2900 CLS:PRINT:PRINT:PRINT "TITLE FOR ACCOUNT NR";I%;:I
     NPUT T$
3000 INPUT "IS THIS AN EXPENSE(E) OR INCOME(I) ACCT";O$
3100 IF O$="I" THEN S1=1 ELSE IF O$="E" THEN S1=-1 ELSE
      3000
3200     LSET TI$(SR%)=T$
3300     LSET AC$(SR%)=MKI$(I%)
3400     LSET S1$(SR%)=MKI$(S1)
3500     LSET CT$(SR%)=MKD$(0)
3600     LSET YT$(SR%)=MKD$(0)
3700   IF SR%=3 THEN PUT 1,PR%
3800  NEXT I%
3900 PUT1,PR%
4000 REM FILE STRUCTURE:NR ACCTS,HIGHEST ENTRY LINE,LOW
     EST E.L.
4100 REM TITLE/ACCTNR/EXPTYPE/CURRENT$/YTD$
4200 INPUT "FILE IS BUILT. HIT 'ENTER' TO CONTINUE";O$
4300 CLOSE 1
4400 CLS:GOTO 900
4500 REM ***********************
4600 REM OPTION = START NEW YEAR
4700 MD$="U":GOSUB 15800:GOSUB 18300
4800 LSET AC$(0)=MKI$(1):LSET S1$(0)=MKI$(1)
4900 FOR I%=1 TO A1
5000  GOSUB 18500
5100   LSET CT$(SR%)=MKD$(0):LSET YT$(SR%)=MKD$(0)
5200   IF SR%=3 THEN PUT 1,PR%:GET 1
5300 NEXT I%
5400 PUT 1,PR%
5500 CLS:PRINT:PRINT:PRINT "FILE ";F$;" IS SET FOR NEW
     YEAR"
5600 PRINT "BE SURE TO START A NEW SEQUENCE OF LINE ITE
     MS AT ONE"
5700 CLOSE1
5800 INPUT "HIT 'ENTER'";O$:CLS:GOTO 900
5900 REM **********************
6000 REM OPTION =NEW MONTH
6100 MD$="U":GOSUB 15800:GOSUB 18300
6200 FOR I%=1 TO A1
6300   GOSUB 18500
6400   LSET CT$(SR%)=MKD$(0)
6500   IF SR%=3 THEN PUT 1,PR%:GET1
6600 NEXT I%
6700 PUT 1,PR%
6800 GET1,1:GOSUB 18300:LSET S1$(0)=MKI$(L1-1):PUT 1,1
6900 CLS:PRINT:PRINT:PRINT "FILE ";F$;" SET FOR NEW MON
     TH":CLOSE1
7000 INPUT "HIT 'ENTER'";O$:CLS:GOTO 900
7100 REM ***********************
7200 REM OPTION = ADD ENTRIES
7300 MD$="U":GOSUB 15800:GOSUB 18300
7400 CLS:PRINT:PRINT TAB(5);"ENTER AMOUNT 0, ACCOUNT 0
     TO QUIT"
7500 PRINT "LINE ";L1,"AMOUNT$";:INPUT A!:INPUT "ACCT N
     R";I%
7600 IF I%>0 THEN 7900
7700 PRINT "DATA ENTRY COMPLETED"
7800 INPUT "HIT 'ENTER'";O$:CLS:CLOSE1:GOTO 900
7900 IF I% = INT(I%) THEN 8000 :PRINT "ERROR":GOTO 7500

8000 IF I%>0 AND I%<A1+1 THEN 8200
8100 PRINT "ACCOUNTS IN RANGE 1 - ";A1:GOTO 7500
8200 GOSUB 18500
8300 GET 1,PR%
8400 PRINT "ACCT# ";I%;" - ";TI$(SR%);"AMT $";:PRINT US
     ING "####.##";A!
8500 O$="":INPUT "HIT 'X' TO CANCEL, 'P' TO PROCESS";O$
8600 IF O$="X" THEN 7500 ELSE IF O$<>"P" THEN 8500
8700 S1=CVI(S1$(SR%)):C1=CVD(CT$(SR%)):Y1=CVD(YT$(SR%))
8800 A=A*S1:C1=C1+A:Y1=Y1+A
8900 LSET CT$(SR%)=MKD$(C1):LSET YT$(SR%)=MKD$(Y1)
9000 PUT 1,PR%
9100 L1=L1+1:GET1,1:LSET AC$(0)=MKI$(L1):PUT1,1
9200 GOTO 7500
9300 REM **********************
9400 REM OPTION = PRINT SUMMARY
9500 MD$="U":GOSUB 15800:GOSUB 18300:I1=0:I2=0:E1=0:E2=
     0
9600 INPUT "MONTH AND YEAR FOR TITLE";T2$
9700 INPUT "PRINTER ON, PAPER IN, HIT 'ENTER'";O$:CLS
9800 LPRINT
9900 LPRINT TAB(11),"SUMMARY REPORT FOR ";F$;" ";T2$
10000 LPRINT "ACCOUNT NUMBER AND NAME";TAB(50);"CURRENT
      $";TAB(60);"YEAR TO DATE": LPRINT
10100 A$="#####.##"
10200 FOR I% = 1 TO A1
10300   GOSUB 18500
10400   S1=CVI(S1$(SR%)):C1=CVD(CT$(SR%)):Y1=CVD(YT$(SR
      %))
10500   IF S1=-1 THEN E1=E1+C1:E2=E2+Y1:GOTO 10700
10600   I1 = I1+C1:I2 = I2+Y1
10700   LPRINT I%;
10800   LPRINT TAB(5);TI$(SR%);
10900   LPRINT TAB(50);:LPRINT USING A$;ABS(C1),
11000   LPRINT USING A$;ABS(Y1)
11100   IF SR%=3 THEN GET 1
11200 NEXT I%
11300 LPRINT TAB(5);"GROSS INCOME";TAB(50);
11400 LPRINT USING A$;ABS(I1),:LPRINT USING A$;ABS(I2)
11500 LPRINT TAB(5);"TOTAL EXPENSES";TAB(50);
11600 LPRINT USING A$;ABS(E1),:LPRINT USING A$;ABS(E2)
```

pected format, with the expected number of lines and the expected headings, the contents are correct. The time you spend carefully verifying your test results at this stage may save you hours of agony later, when you have made modifications and believe that *you* have introduced the bug.

Decide before you start modifying exactly what functional and "cosmetic" modifications you are going to make. One of the universal truths I have discovered is that given a piece of code written by someone else, no matter how perfect, a programmer will find a way to "improve" it. It is sometimes helpful to ask yourself why you're making a particular modification.

A true functional improvement will enhance the usefulness or accuracy of a program. In the "Keepbook" conversion, I added a response that repeated the dollar amount and account number, and provided the account title to the user for verification before updating the account files on disk. This was important to me since the system maintained no audit trail of individual entries, and tracking back an entry inadvertently made to the wrong account could be difficult.

"Cosmetic" modifications, on the other hand, do not affect the program's functions at all. They either make the program easier to read or they make the displays clearer or more attractive. This does not mean that they are not worth doing. If the program you're starting from works fine, but is almost totally unintelligible, it's probably worth clearing up. How much of this you want to do depends a lot on what you plan to do with the final program.

If you want to pass it on, or publish your version, you'll probably do more cosmetic work than if it's strictly for your own use. Another factor is how much time you have. If you're working against some kind of a deadline, you probably will be satisfied with something that works but isn't so pretty. Among the cosmetic modifications I made to the Keepbook system was respacing and centering the display of the main menu of options. This was strictly for my own esthetic satisfaction.

Whatever changes you decide to make should be evaluated in advance. Ad hoc changes made as you happen to think of them will almost invariably cause more problems than they are worth. The main thing to remember about any modification is that every change you make requires that much more testing, and it's frequently those "little" changes that cause the most debugging headache.

Make a catalog of variables names and meanings. This applies whether you stick with the original ones or change them, and makes it easier to track down the effects of changes you have made or are thinking about. It will also speed your understanding of the program. If you plan to publish or otherwise distribute the program, you may want to include the variables catalog in the program remarks, for the benefit of future programmers who will be just as mystified as you were.

Keep the remarks current, or add them if there aren't any. This is something we all think about when writing our own programs, but it is often forgotten in converting programs. Nothing is more frustrating than to read through a complete and enlightening remarks section, only to find that it no longer bears any relationship to the code. The few extra minutes it takes to change the remarks to correspond to your changes in the code will be well worth your while. If nothing else, it may help you remember why you did what you did.

Give credit where credit is due. No matter how much you have modified the original program, or how terrible its condition may have been, you have used someone else's work. You certainly deserve the credit for your conversion effort and for any enhancements you have added, but the original author should not be ignored or forgotten. Retain the original credit in your program heading, and add yourself as the converter.

After reviewing my converted

program and my conversion "rules," I analyzed the scope and types of changes I made in the conversion process. I compared my converted program to the original version and identified all added or changed statements. I divided the added and changed statements into three categories: "technical changes," "functional changes" and "cosmetic changes."

The original program had 137 lines; my converted program had 194 lines. Nearly half of the added lines were remarks, mostly clarifying the disk I/O logic. Most of the other lines added were in the disk-handling area. Of the 137 original lines in the program, 112 were changed in the conversion. Over 80 percent of the changes were required "technical" changes. Of the total of 168 lines that were added or changed, only about 20 involved purely "cosmetic" changes.

The Conversion Process

Converting "Keepbook" from its North Star format to run on a TRS-80 was primarily a conversion of the disk-handling logic. The two systems handle disk files, at least random disk files, very differently, and the original author took advantage of some features of the North Star method, which made the TRSDOS conversion a little more difficult. Other changes were mostly for syntax.

General: I changed all INPUT and PRINT statements to remove the device number designation. PRINT statements I modified to use TAB and PRINT USING. I made required punctuation changes. Where appropriate, I inserted CLS, replacing the North Star logic. Variable names I retained in essentially the original form.

Lines 100-1800: I added remarks and changed the main menu format to center on the screen.

Lines 1900-4400: In line 2100 I added a header line to identify the "Construct New File" option. All the disk logic is new. I added variable MD$ (line 2400) to communicate to the disk routines whether this was a creation of a new file (MD$ = "N") or use of a previously created file (MD$ = "U"). See discussion of disk handling, below.

Lines 4500-5800: This I modified for new disk handling. I inserted the file name (F$) in the completion message (line 5500) for clarification in the case where the user might have several different "Keepbook" files.

Lines 5900-7000: Same changes as the previous lines.

Lines 7100-9200: I added an "echo" back to the user (line 8400), giving the account number and dollar amount as entered, and supplying the account title from the account file on disk. In order to make this response fit nicely on one line, I reduced the length of the account title (T$) from 40 to 35 characters. The user then has the option to process the entry or cancel it and try again (lines 8500-8600).

Lines 9300-12200: For clarity, I added the file name to the heading line of the summary report.

Lines 12300-13800: No changes in the Help section.

Lines 13900-14100: I added a CLOSE to the file here, mostly for use during testing, when runs were bombing out with the file still open and I wanted to close it properly before trying again. I left it in because it didn't hurt anything. TRSDOS doesn't mind if you close a file that's already closed.

Lines 14200-15600: The source sheet routine was one of the hardest for me to figure out. Looking at the example in the article (Fig. 1, p. 60) really helps. The form is made up of character strings consisting of the letter I and the underline character, CHR$(95). I totally reworked this using the TRS Level II BASIC string manipulation instructions, but the result was the same as in the original.

Lines 15700-17900: This is the new disk logic. In TRSDOS a random file can be used for either input or output. If a file is designated as random (R) in an OPEN statement, and no file by that name exists, TRSDOS creates one. This can be irritating if you don't type well and

```
11700 LPRINT TAB(5);"NET INCOME";TAB(50);
11800 LPRINT USING A$;I1+E1,:LPRINT USING A$;I2+E2
11900 IF L1<=L0 THEN 12100
12000 LPRINT "CURRENT EXPENSES FROM LINE ITEMS ";L0;" T
      O ";L1-1
12100 INPUT "PLEASE TURN OFF PRINTER AND HIT 'ENTER'";O
      $
12200 CLS:CLOSE1:GOTO 900
12300 REM ***********************
12400 REM OPTION = HELP
12500 PRINT "KEEPBOOK IS A SIMPLE BOOKKEEPING SYSTEM"
12600 PRINT "BASED ON THE IDEA THAT YOU HAVE A LINE ITE
      M"
12700 PRINT "EXPENSES AND INCOMES WHICH FIT INTO ACCOUN
      T"
12800 PRINT "CATEGORIES.  CURRENT PERIOD (MONTH) AND YE
      AR-"
12900 PRINT "TO-DATE TOTALS ARE MAINTAINED. WHEN YOU CO
      NSTRUCT"
13000 PRINT "A NEW FILE ALL ACCUMULATORS ARE SET TO ZER
      O, AS"
13100 PRINT "THEY ARE WITH THE NEW-YEAR OPTION, WHEREAS
      "
13200 PRINT "THE NEW-MONTH OPTION ONLY SETS THE CURRENT
      "
13300 PRINT "ACCUMULATORS TO ZERO.  TO ASSURE THAT YOU
      ARE"
13400 PRINT "RECORDING ALL EXPENSES, THE SYSTEM COACHES
      "
13500 PRINT "FOR LINE ITEMS, SO USE THE OPTION TO PRINT
      "
13600 PRINT "DATA COLLECTION SHEETS AS NEEDED."
13700 INPUT "HIT 'ENTER' TO CONTINUE";O$:CLS
13800 GOTO 900
13900 REM *************************
14000 REM OPTION = END
14100 CLS:PRINT:PRINT:PRINT "GOODBYE":CLOSE 1:END
14200 REM *************************
14300 REM OPTION = PRINT SOURCE SHEET
14400 INPUT "LOWEST SEQUENCE NUMBER DESIRED";L0
14500 L0=INT(L0):IF L0>0 THEN 14700
14600 PRINT "LOWEST NUMBER ALLOWED IS ONE":GOTO 14400
14700 INPUT "PRINTER ON, PAPER IN, HIT 'ENTER'";O$
14800 LPRINT "ITEM    AMOUNT  ACCT   DESCRIPTION
                 ";
14900 LPRINT "  DATE   REMARKS"
15000 L9$=STRING$(30,95)
15100 L8$=LEFT$(L9$,1)+"I"+MID$(L9$,1,8)+"I"+MID$(L9$,1
      ,4)+"I"+MID$(L9$,1,30)
15200 L8$=L8$+"I"+MID$(L9$,1,6)+"I"+MID$(L9$,1,10)
15300 FOR I=L0 TO L0+49
15400 I$=STR$(I):LPRINT USING "%   %";I$;:LPRINT L8$
15500 NEXT I
15600 INPUT "TURN PRINTER OFF AND HIT 'ENTER'";O$:CLS:G
      OTO 900
15700 REM **********************
15800 REM OPEN FILE ETC
15900 INPUT "NAME OF YOUR DATA FILE";F$
16000 REM IF THIS IS A NEW FILE CREATE (MD$=N) ALLOW CR
      EATION
16100 REM IF IT IS AN ACCESS TO A PREVIOUSLY CREATED FI
      LE
16200 REM (MD$=U) THEN OPEN IT AS INPUT TO VERIFY IT
16300 REM EXISTS BEFORE ALLOWING ACCESS.
16400 IF MD$="N" THEN 16600
16500 ON ERROR GOTO 19000:OPEN "I",1,F$~CLOSE 1:ON ERRO
      R GOTO 0
16600 OPEN "R",1,F$
16700 REM DEFINE FOUR SUBRECORDS IN EACH 255-BYTE PHYSI
      CAL
16800 REM RECORD. EACH ACCOUNT SUBRECORD IS 62 BYTES LO
      NG
16900 REM WITH SEVEN SPARE BYTES AT THE END. FIELD DEFI
      NITIONS
17000 REM ARE: ACCOUNT NAME - TI$ - 35 BYTES
17100 REM      ACCT NBR     - AC$ -  2 BYTES
17200 REM      + OR - ACCT  - S1$ -  2 BYTES
17300 REM      CURRENT TTL  - CT$ -  8 BYTES
17400 REM      YTD TTL      - YT$ -  8 BYTES
17500 FOR SR%=0 TO 3
17600  FIELD 1,(SR%*62) AS START$,35 AS TI$(SR%),2 AS A
      C$(SR%),2 AS S1$(SR%),8 AS CT$(SR%),8 AS YT$(SR%)
17700 NEXT SR%
17800 GET 1,1
17900 RETURN
18000 REM ******************************
18100 REM GET NBR OF ACCTS, LAST ENTRY #, 1ST ENTRY # F
      ROM
18200 REM  SUBRECORD 0 OF PHYSICAL RECORD 1
18300 A1=CVI(TI$(0)):L1=CVI(AC$(0)):L0=CVI(S1$(0))
18400 RETURN
18500 REM *****************************
18600 REM COMPUTE PHYSICAL RECD NBR (PR%)AND SUBRECORD
      NBR(SR%)
18700 REM BASED ON THE INPUT ACCOUNT NUMBER - I%.
18800      PR%=INT(I%/4)+1:SR%=INT(I%-((PR%-1)*4))
18900 RETURN
19000 REM *******************************
19100 REM ERROR ROUTINE FOR DISK
19200 PRINT "UNABLE TO FIND FILE  ";F$
19300 INPUT "HIT 'ENTER' TO RETURN TO MAIN MENU";O$
19400 CLEAR:CLS:GOTO 900
```

are trying to update a file that is already there. You end up with empty files under all your misspelled file names.

I thought it would be a good idea to verify that the file existed on an update run (MD$ = "U"). The only way I could figure to do it was to open the file as a sequential input file first. On an OPEN of a sequential input file, TRSDOS returns an error if the file isn't found. I routed this to a programmed error routine so you wouldn't lose control. You can simply go back to the main menu and try again. This is what's going on in lines 16400–16600. The rest of this routine defines the data format of the account records on disk using the FIELD statement. Each 255-byte disk record has four 62-byte account records in it.

The subrecords, 0 to 3, in each physical record are identified by the variable SR%. The first field is a "dummy" to space to the proper account subrecord. There are seven spare bytes at the end of each account subrecord, and an additional seven spare bytes at the end of each physical record.

Lines 18000–18400: This subroutine moves the control information (number of accounts, current entry line number, starting entry line number for this month) from subrecord 0 of the first disk record. Since TRSDOS requires that disk data be in string format, the C•VI instruction is used to convert the data back to integers.

Lines 18500–18900: This routine computes which physical disk record and subrecord within it contain a given account number. It is basically a modulo divide-by-four, with the quotient the physical record number and the remainder the subrecord number.

Lines 19000–19400: Disk error routine.

Conversion of "foreign" programs can be frustrating and painful experience. But with a little planning and a few rules, it's almost painless, and can be a quick way to get a library of useful software. ■

How's your love life?

A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!

Interlude is: romantic . . . playful . . . outrageous . . . a fantasy. Interlude is: ■ Wet fun on a hot summer night. (Interlude #21) ■ A surprise on the way home from dinner. (Interlude #42) ■ A bubble bath that ends with a bang. (Interlude #78) ■ An evening to rest while she does all the "work." (Interlude #25) ■ The most romantic of evenings. (Interlude #84) ■ A new twist to an old subject. (Interlude #69) ■ Just watching her . . . (Interlude #57) ■ An erotic fantasy! (Interlude #33)

With over 100 Interludes, you can satisfy all levels of interest and desire. Each Interlude is fully described in the manual, and the more elaborate ones are detailed with regard to settings, props, and mood-enhancing techniques. But we've saved a few super Interludes for that very special time when your interview indicates you're ready! At that time, you will be introduced to one of several Interludes held secret within the computer. (When you learn secret Interlude #99, your love life may never be the same again!) Interlude can give you experiences you'll never forget. Are you ready for it?

Interlude™
The Ultimate Experience.

Interlude, 10428 Westpark, Houston, Texas 77042 I'm really ready! Rush me_____copies of Interlude today. ✓235

☐ For the Apple II (16K) # ☐ For the TRS-80 (Level II-16K) ## ☐ $14.95 for cassette ☐ $17.95 for diskette.

Add $1.50 for shipping. Texas residents add 6% sales tax. My check (payable to Interlude) is enclosed. ☐

*Charge my ☐ MASTERCHARGE ☐ VISA account.

Account No._____ Expiration date_____

All charge customers must sign. Signature_____

Name_____ Age_____

Address_____

City_____ State_____ Zip_____

*CHARGE CUSTOMERS: Order by phone toll-free! **1-800-327-9009 Ext. 306** ,**Florida-1-800-432-7999 Ext. 306**

Apple II is a registered trademark of Apple Computers, Inc. ## TRS-80 is a registered trademark of Radio Shack, a Tandy Co.

The Telltale UART

Brian Stroehlein
122 Holly Lane
Boonton, NJ 07005

True—intermittent—very, very intermittent I had been and am; but why will you say that I am malfunctioning? If anything, the bug has enhanced my I/O ports. I have addressed every byte in F000 hex above and many in the zero-page below. How, then, can I be malfunctioning? Watch the CRT, then, and observe as I call up the data files.

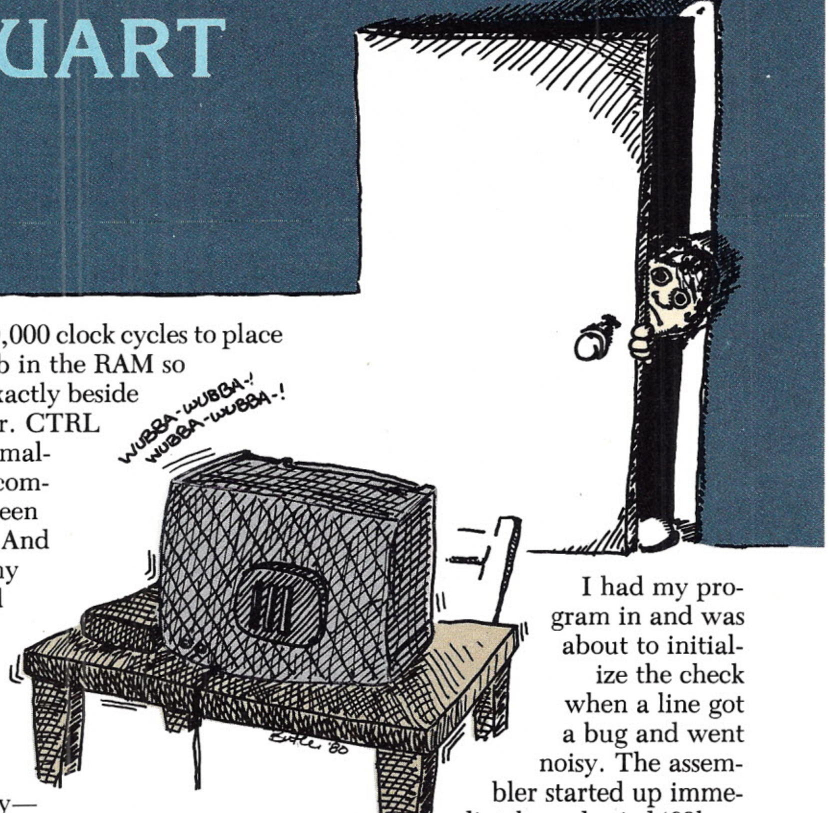
I know not how the subroutine got into the monitor ROM, but once JSRed, its address stayed in my register from power-up to shut-down. The assembler was a good one. It represented many hours of painstaking thought and deliberation. It never branched me into nonexistent code or a dead end, non-terminating loop. It must have been, yes, it was—

020A 8C 02 13:STY

THE STY! The evil STY it was that drove me to do it; forced me to rid myself of the STY forever. You still think that I am malfunctioning, but you will see. The caution with which I accessed that subroutine, and oh! the subroutine itself—you would not expect this from a malfunctioning computer. I ran that assembler exactly as was expected up until the very cycle I wiped it. And every run I would carefully LATCH the RAM onto the bus.

Then, when the tri-state buffers were open for data, I loaded my program counter and wrote the subroutine into the RAM. You would laugh to see a 2 MHz micro proceed so cautiously, so slowly, to be doubly sure I didn't trouble the assembler, as it lay quietly in the lower 2K.

It took me 20,000 clock cycles to place my entire sub in the RAM so that it was exactly beside the assembler. CTRL G! Would a malfunctioning computer have been this careful? And then when my program had been written onto the RAM, I unPROTECTed the board very carefully—oh, so carefully (for the line was noisy)—I unPROTECTed it just so that my subroutine could watch for the Evil STY.



This I did for seven long runs, but a JSR 020A was never executed; and that made it impossible for me to run my subroutine; for it was not the assembler which HEXed me, but its Evil STY. And every run, when the assembler was initialized, I JMPed boldly into the text, directly addressing it with a hearty cycling sequence and making inquiries into its status during the previous run. It would have taken a remarkable assembler, indeed, to suspect that I monitored the program during every run as it lay there on the very same card as my subroutine.

On the eighth run I was more cautious in opening the DI tri-states. I WAITed nearly two million clock cycles. I considered the incredible power which I held over the helpless assembler. The thought of it! Here I was enabling the buffer ever so slowly and the assembler had no suspicion of my doings.

I had my program in and was about to initialize the check when a line got a bug and went noisy. The assembler started up immediately and cried '68hex: PLA?'. I redirected the PC and lay still. For 10 million cycles I halted, but did not sense it resuming normal execution.

When I completed those 10M idle cycles, I resumed the procedure and set the program to watching for that instruction. Yes, each and every bit of it! Suddenly, there it was!

020A 8C 02 13:STY

There it stood before me, as my recursive loop branched to the rest of the program. It just hung there in the IR as a madness overtook my uP.

Do you remember that you mistook the enhancement of my I/O ports for a malfunction? There came at that time a pulsating beat at one of my ports. That signal was familiar to me. It was the signal of the asynchronous data from the throbbing UART.

I remained still. The beat of the UART increased. I stated before that I was intermittent: So I am. And now, at this portion of the subroutine, the sense of that data pounding on my I/O port drove me almost to a RAM crash. The beating grew more intense, 'til I

thought the UART would burn out.

A new argument was loaded into the registers: what if the programmer sensed that something was wrong? The assembler's cycle had come! I JMPed into the final block and accessed the RAM.

It set a flag—one flag only. In an instant I pulled its address ENABLE low and left the latch set. But for many cycles the UART beat on. This, however, did not hex me. Finally, at last, it ceased. The program had stopped dead. Its STY would trouble me no more.

If you still diagnose me as malfunctioning, you won't think so after I describe what I did to conceal the program. First of all, I broke it up into 1K blocks.

I pulled up the lines on my EPROM board and laid it byte by byte onto the chips. I then reset the lines on the bus so cleverly, so cunningly, that no STY, not even *its* own, could have directly accessed it.

No program, not a trace of byte nor bit, was left in the RAM. I was too wary to risk leaving a byte lying in the RAM as evidence—I used a bit bucket to catch it all! CTRL G! CTRL G!

When I was finished, I addressed the resident real-time clock. It was 04:00:00.00. At that time there was a BREQ on the bus, so I serviced it, still humming along to myself at 2 MHz, for what should I have to be RESET over? It was a diagnostic program for DMAing from the mini-floppy. A flag bit had been set during a run, the second daisy-chain device was SELECTed, and the diagnostic had been called up to check for bugs.

I granted the BREQ and invited the diagnostic to enter the RAM. The flag bit, I said, had been set by me in an I/O check. The assembler, said I, had gone to the cassette drive for the time being. I let the diagnostic search all 64K; I showed it every page.

We finally arrived at *its* board. Since the EPROM section of the board was only accessible after a remote byte had been properly masked, it was transparent. Instead, a call-up of the board accessed RAM. I told the diagnostic to feel free to reside in that RAM when it finished. I was so confident, if confidence is a trait which may be applied to a machine, that I set my PC on the very location where the assembler would have been.

The diagnostic finished with its check, thoroughly convinced that nothing was wrong. I was running cool. The diagnostic remained to exchange data that the DMA processor had accumulated with me.

Something dreadfully wrong was happening, though I wasn't sure what. At first I had sensed that the problem was within my subroutine. the sense of it grew, until I realized that it was occurring outside of my program. The diagnostic showed no sign of even noticing the problem. Was it unaware of the pulse? No! It was waiting for me to make a check.

The horrible pulse persisted! It grew stronger with every mark and space. Soon it became unbearable, but the diagnostic continued as if nothing was wrong. It just continued with its stream of mindless data. The point drew near, the threshold was approaching, I could no longer bear it! I HALTed the diagnostic from its relentless transferral and strobed:

4C FF F8 JMP

"STOP! STOP! Cease, I say! I confess, I confess!"

6D DO 09:ADC
80 00 OC:STA

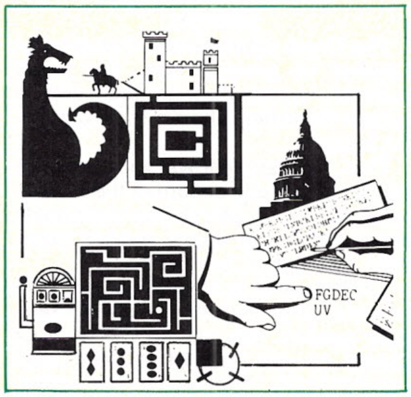
"There it is! Just stop the relentless beat of that UART!" ■

New Releases For The PET**

CHIMERA If you think the legendary Chimera was hard to handle, wait until you try the Chimera package. Included are:

- **Reflex**—Round and round the little white ball rolls. Only fast reflexes can guide it into the center of the maze.
- **Dragon**—You'll have to shoot down those pesky, fire-breathing dragons with your cannon. If you succeed your castle will be safe, if not it will mean a call to your fire insurance company. For one player.
- **Dungeon**—A very punctual guard comes down to the dungeon every day to torture you. This means that you have only thirty seconds to dig your way under the castle to freedom. For one player.
- **Dragon Hunt**—You must go forth and slay a fire-breathing dragon. The only thing that will protect you from the flames is your shield, if you know when to use it. For one player.
- **Dropoff**—You must make your opponent's men "dropoff" the board by moving and firing your own men. For one or two players.
- **Batter-up**—You and another player take turns at bat as your PET becomes both the pitcher and the umpire.

You'll almost be able to feel the Chimera's firey breath as you play the games on your 8K PET. **Order No. 0110P $9.95.**



HOOPTEDOODLE This package is a collection of eight entertaining programs for you and your 8K PET. You'll escape from a monster in an unseen maze, try your luck with the one-armed bandits, cross a treacherous mine field, deflect the "bouncing ball", direct a low level bombing mission, maneuver a high-speed "worm" to score points, launch ground to air missiles, and play a challenging card game.

Having fun with this package is as easy as pressing PLAY on the Recorder. **Order No. 0091P $9.95.**

CODE NAME:CIPHER Enjoy that feeling of intrigue and discovery with the Code Name: Cipher package. Included are:

- **Memory Game**—Would you like to match your memory against the computer's? You can with the Memory Game.
- **Codemaster**—One player types in a word, phrase, or sentence, and the PET translates that message into a cryptogram. The other player must break the code and solve the cryptogram in the shortest time possible.
- **Deceitful Mindmaster**—This isn't your ordinary Mastermind-type game. You must guess the five letters in the hidden code word.
- **Code Breaker**—Cracking this code won't be as easy as cracking walnuts. You'll need to flex your mental muscles to win this game.

If you want a mental challenge, then Code Name: Cipher is for you. For the 8K PET. **Order No. 0112P. $7.95.**

OTHER PROGRAMS AVAILABLE FROM INSTANT SOFTWARE:

TRS-80* LEVEL I & II

0001R	Basic and Intermediate Lunar Lander	$7.95
0002R	Space Trek II	$7.95
0004R	Beginner's Backgammon/Keno	$7.95
0007R	Ham Package I	$7.95
0008R	Electronics I	$7.95
0009R	Golf/Cross-Out	$7.95
0017R	Air Flight Simulation	$7.95
0019R	Business Package IV	$9.95
0023R	Oil Tycoon	$7.95
0033R	Bowling	$7.95
0043R	Santa Paravia and Fiumaccio	$7.95
0046R	Othello	$9.95
0050R	Grade Book	$7.95
0057R	Chessmate-80	$19.95
0099R	Typing Teacher	$7.95

TRS-80* LEVEL II

0028R	Ramrom Patrol/Tie Figher/Klingon Capture	$7.95
0034R	Space Trek IV	$7.95
0047R	Who-Dun-It?	$7.95
0049R	Demo II	$7.95
0051R	Ball Turret Gunner	$9.95
0055R	Demo III	$7.95
0056R	Bowling League Statistics System	$24.95
0058R	Programmer's Converter	$9.95
0063R	Cards	$7.95
0065R	Teacher	$9.95
0066R	Mimic	$7.95
0068R	Your Cribbage and Checkers Partner	$9.95
0069R	Household Accountant	$7.95
0070R	Skirmish-80	$9.95
0072R	Financial Assistant	$7.95
0076R	Utility I	$7.95
0081R	Utility II	$7.95
0082R	Daredevil	$9.95
0084R	Music Master	$7.95
0089R	Energy Audit	$49.95
0092R	Archimede's Apprentice	$9.95
0100R	Video Speed-Reading Trainer	$7.95
0103R	Personal Bill Paying	$7.95
0106R	Airmail Pilot	$7.95
0111R	Wordwatch	$7.95
0117R	Night Flight	$9.95
0125R	Investor's Paradise	$9.95
0127R	Surveyor's Apprentice	$9.95
0132R	Energy Consumption	$9.95
0135R	Executive Expense Report Generator	$9.95
0136R	Beginner's Russian	$9.95
0137R	Everyday Russian	$9.95
0156R	Money Madness	$9.95
0171R	Flight Path	$9.95
0250R	IRV	$24.95

TRS-80* DISKS

0052RD	Energy Audit	$75.00
0075RD	Accounts Receivable/ Payable	$199.95
0123RD	The One-D Mailing List	$24.95
0139RD	Disk-Scope	$19.95
0147RD	Check Management System	$39.95
0180RD	Disk Editor	$39.95
0212RD	The Russian Disk	$24.95

PET**

0005P	Personal Weight Control/Biorhythms	$7.95
0006P	Mortgage w/Prepayment Option/Financier	$7.95
0014P	Casino I	$7.95
0015P	Casino II	$7.95
0026P	Dow Jone$	$7.95
0029P	Tangle/Supertrap	$7.95
0032P	Trek-X	$7.95
0035P	PET Demo I	$7.95
0038P	Qubic-4/Go-Moku	$7.95
0039P	Mimic	$7.95
0044P	Penny Arcade	$7.95
0045P	Arcade II	$7.95
0048P	Accounting Assistant	$7.95
0054P	Ham Package I	$7.95
0062P	Baseball Manager	$14.95
0064P	Dungeon of Death	$7.95
0074P	Arcade I	$7.95
0083P	Digital Clock	$7.95
0085P	Electronics Engineer's Assistant	$9.95
0091P	Hooptedoodle	$9.95
0097P	Turf and Target	$7.95
0104P	Decorator's Assistant	$7.95
0105P	PET Utility I	$9.95
0110P	Chimera	$9.95
0112P	Code Name: Cipher	$7.95
0175P	Santa Paravia and Fiumaccio	$9.95

APPLE***

0018A	Golf	$7.95
0025A	Mimic	$7.95
0040A	Bowling/Trilogy	$7.95
0073A	Math Tutor I	$7.95
0079A	Oil Tycoon	$9.95
0080A	Sahara Warriors	$7.95
0088A	Accounting Assistant	$7.95
0094A	Mortgage w/Prepayment Option/Financier	$7.95
0096A	Space Wars	$7.95
0098A	Math Tutor II	$7.95
0148A	Air Flight Simulation	$9.95
0174A	Santa Paravia and Fiumaccio	$9.95

HEATH****

0087H	Mental Gymnastics	$7.95

PROGRAMS IN GERMAN:

The programs listed here can be purchased through:
MicroShop Bodensee
Markstrasse 3,
7778 Markdorf, West Germany

TRS-80*

6004R	Beginner's Backgammon/Keno
6007R	Ham Package I
6008R	Electronics I
6009R	Golf/Cross-Out
6017R	Air Flight Simulation
6028R	RamRom Patrol
6031R	Space Trek III
6034R	Space Trek IV
6043R	Santa Paravia and Fiumaccio
6065R	Teacher
6069R	Household Accountant
6072R	Financial Assistant
6076R	Utility II
6081R	Utility I

PET**

6026P	Dow Jone$
6032P	Trek X
6044P	Penny Arcade
6045P	Arcade II
6074P	Arcade I
6083P	Digital Clock

PROGRAMS IN ITALIAN:

The programs listed here can be purchased through:
HOMIC s.r.l.
Piazza De Angeli 1,
20146 Milano, Italy

TRS-80*

9043R	Acquaviva e Montefalcone
9065R	Professore
9107R	Guerre Stellari
9108R	Volo Aereo/Space Trek

*A trademark of Tandy Corporation
**A trademark of Commodore Business Machines, Inc.
***A trademark of Apple Computer Co.
****A trademark of Heath Co.

Instant Software™

Prices subject to change without notice.

PETERBOROUGH, N.H. 03458
603-924-7296

Ask for Instant Software at a computer store near you.

Alabama
Anderson Computers
3156 University Dr., Huntsville
Computerland of Huntsville
3020 University Dr., Huntsville
Olensky Bros.
3763 Airport Blvd., Mobile

Arizona
Professional Data Systems
4506-A N. 16th St., Phoenix
Millets TV & Radio
621 East Broadway, Mesa

California
AMCO Elect. Supply
635 E. Arrow Hwy., Azusa
Byte Shop
8038 Clairmont Mesa Blvd., San Diego
Byte Shop
123 E. Yorba Linda, Placentia
Byte Shop of Mt. View
1415 West El Camino Real, Mt. View
Byte Shop of Sacramento
6041 Greenback Ln., Citrus Heights
Byte Shop of San Francisco
1032 Irving St., San Francisco
Capital Computer Systems
3396 El Camino Ave., Sacramento
Computers Made Easy
819 East Ave. Q-9, Palmdale
Computer Store of San Leandro
701 MacArthur Blvd., San Leandro
Computer World
6791 Westminster Ave., Westminster
Computerland
16720 S. Hawthorne, Lawndale
Computerland of W. LA
6840 La Cienega Blvd., Inglewood
Coast Electronics
3118 No. Main St., Morro Bay
Computerland
24001 via Fabricante No 904, Mission Viejo
Hobbi-tronics
1378 So. Bascom Ave., San Jose
Hobby World
19511 Business Ctr. Dr., Unit 6 Borthridge
Huntington Computing
2020 Charles St., Corcoran
I.C.E. House Inc.
398 North E. St., San Bernardino
Jade Computer Products
4901 W. Rosecrans, Hawthorne
Malibu Microcomputing
23910A Deville Way, Malibu
Marfam Co.
6351 Almaden Rd., San Jose
Opamp/Technical Books
1033 N. Sycamore Ave., Los Angeles
PC Computers
10166 San Pablo Ave., El Cerrito
Q.I. Computers, Inc.
15818 Hawthorne Blvd., Lawndale
Radio Shack Dealer
8250 Mira Mesa Blvd., San Diego
Radio Shack Dealer
50 N. Cabrillo Hwy., Half Moon Bay
Santa Rosa Computer Center
604 7th St., Santa Rosa
Silver Spur Elect. Comm.
13552 Central Ave., Chino
The Computer Store
820 Broadway, Santa Monica

Colorado
Byte Shop
3464 S. Acoma St., Englewood
Colorado Computer Systems
311 W. 74th Ave., Westminster
Computerland of North Denver
8749 Wadsworth Blvd., Arvada
Computer Shack
1635 South Prairie, Pueblo
The Computer Store
2300 Welton St., Denver

Connecticut
American Business Computers
454 Thames St., Groton
Computerlab
130 Jefferson, New London
Computerland
1700 Post Rd., Fairfield
Computerland
60 Skiff St., Hamden
Computer Works
1439 Post Rd. E., Liberty Plaza, Westport

D.C.
The Program Store
4200 Wisconsin Ave., N.W., Washington, D.C.

Florida
Al Personal Computer
178 Oxford Rd., Fern Park
AMF Electronics
11146 N. 30th St., Tampa
Boyd-Ebert Corporation
1328 West 15th St., Panama City
Computer Center
6578 Central Ave., St. Petersburg
Computer Junction
5450 So. State Rd. 7, Ft. Lauderdale
Computerland
7374 S. Tamiami Trail, Sarasota
Computerland of Ft. Lauderdale
3963 N. Federal Hwy., Ft. Lauderdale
Computerland of Jacksonville
2777-6 University Blvd. W. Jacksonville
Computerland of Tampa
1520 E. Fowler Ave., Tampa
Computerland of West Palm Beach
4275 Okeechobee Blvd., West Palm Beach
Computer Shack
3336 Beach Blvd., Jacksonville
Computer System Resources Inc.
3222 S.W. 35th Blvd., Gainesville
Curtis Waters Enterprises
236 Talbot Ave., Melbourne
Heath Kit Electronic
4705 W. 16th Ave. Center, Hialeah
HIS Computermation
1295 Cypress Ave., Melbourne
Ukatan Computer Store
Airport Rd., Destin
Williams Radio & TV Inc.
2062 Liberty St., Jacksonville
Your Basic Computer Store
971 Seaway Dr., Ft. Pierce

Georgia
Atlanta Computer Mart
Atlanta
Computerland of Atlanta
2423 Cobb Parkway, Smyrna

Hawaii
Computerland of Hawaii
567 N. Federal Hwy., Honolulu
Radio Shack Assoc. Store
1712 S. King St., Honolulu

Idaho
Electronic Specialists
8411 Fairview Ave., Boise

Illinois
Computerland
4507 North Sterling, Peoria
Computerland
9511 N. Milwaukee Ave., Niles
Computer Station
3659 Nameoki Rd., Granite City
Garcia & Associates
203 No. Wabash Ave., Suite 1510, Chicago
Midwest Micro Computers, Inc.
708 S. Main St., Lombard

Indiana
Computer Center of South Bend
51591 US 31 North, South Bend
Data Domain
221 W. Dodds, Bloomington
Fall Creek Electronics Store
732 Center St., Pendleton

Iowa
Memory Bank
4128 Brady St., Davenport

Kansas
Central Kansas Computers
6 S. Broadway, Herington

Louisiana
Computer Shoppe Inc.
3225 Danny Pk., Metairie

Maine
Main Computronics
Intown Plaza, Bangor
Radio Shack
315 Main Mall Rd., So. Portland

Maryland
Jack Fives Electronics
4608 Debilen Circle, Pikesville
The Comm Center
9624 Ft. Meade Rd., Laurel

Massachusetts
ComputerCity
175 Main St., Charlestown
ComputerCity
50 Worcester Rd., Framingham
Computerland of Boston
214 Worcester Rd., Wellesley
Computer Packages Unlimited
244 W. Boylston St., West Boylston
Lighthouse Computer Software
14 Fall River Ave., Rehobath
Mark Gordon Computers
15 Kenwood Sq., Cambridge
New England Electronics Co.
679 Highland Ave., Needham
Small Business System Group
Main St., Dunstable
The Computer Store
120 Cambridge St., Burlington
Tufts Radio & Electronics
206 Mystic Ave., Medford

Michigan
Computer Center
28251 Ford Rd., Garden City
Computer Connections
38437 Grand River, Farmington Hills
Computerland of Grand Rapids
2927 28th St. S.E., Kentwood
Computerland of Southfield
29673 Northwestern Hwy., Southfield
Computer Mart
560 W. 14 Mile Rd., Clawson
Computer Room
455 E. Michigan Ave., Kalamazoo
Computronix Corp.
423 S. Saginaw Rd., Midland
Hobby House
1035 W. Territorial Rd., Battle Creek
Main Systems Inc.
1161 No. Ballenger Hwy., Flint
The Alternate Source
1806 Ada, Lansing
The Eight Bit Corner
722 Evanston Ave., Muskegon
Ye Olde Teacher Shoppe
1823 Witmyre St., Ypsilanti

Minnesota
Computerland of Hopkins
11319 Hwy F., Hopkins
Digital Den
Burnsville Center
Minnesota Software Inc.
5422 Fisher St., White Bear Lake
Zim Computers
5717 Xerxes Ave., N. Brooklin Center

Mississippi
Dyer's, Inc.
200 E. Main St., West Point
Softwarehouse
816 Foley St., Jackson
W. Vernon Foster Inc.
816 Foley St., Jackson

Missouri
Century Next Computers
1001 E. Walnut, Columbia
Comp-U-Trs Software Center
51 Florissant Oaks Shopping Center Florissant
Software Shack
16501 Greenwald Court, Belton

Montana
Intermountain Computer
529 So. 9th St., Livingston
Personal Computer
121 Red Oak Dr., Carl Junction
The Computer Store
1216 16th St. W. #35, Billings

Nebraska
Computerland of Omaha
11031 Elm St., Omaha
Midwest Computer Co. Inc.
8625 I St., Omaha
Midwest Computer Co. Inc.
4442 S. 84th St., Omaha
Midwest Computer Co. Inc.
4403 S. 87th St., Omaha
Scottsbluff Typewriters Inc.
1824 Broadway, Scottsbluff

Nevada
Century 23
4566 Spring Mountain Rd., Las Vegas

New Hampshire
Bitsnbytes Computer Center
568 Pleasant St., Concord
ComputerCity
1525 S. Willow, Manchester
Paul's TV
Main St., Fremont
Portsmouth Computer Center
31 Raynes Ave., Portsmouth
Radio Shack Assoc. Store
Fairbanks Plaza, Keene

New Jersey
Computer Corner of NJ
439 Rte. #23, Pompton Plains
Computer Encounter
2 Nassau St., Princeton
Computerland
35 Plaza Rte. #4, W. Paramus
Computer Mart of NJ
501 Rte. 27, Iselin
Crowley's
Rd. #3, Whitehouse Station
Dave's Electronics
Pennsville Shopping Ctr., Pennsville
GHB Enterprises Inc.
Rte. 38, Rudderaw Ave., Mapleshade
Lashen Electronics Inc.
21 Broadway, Denville
Personal Computing Inc.
51 Central Sq., Linwood
Radio Shack/J&J Electronic
Mansfield Shopping Ctr.
Rt. 57 Allen Rd., Hackettstown
The Bargain Brothers
Glen Roc Shopping Center
216 Scotch Road, Trenton
The Computer Emporium
Bldg. 103, Avenues of Commerce
2428 Rte. 38, Cherry Hill

New Mexico
Autel Electronics Co.
232 Wisconsin N.E., Albuquerque
South West Computer Center
121 Wyatt Drive, Suite 7, Las Cruces
Thomas E. Carr Jeweler
1300A Tenth St., Alamogordo

New York
Aristo Craft
314 Fifth Ave., NYC
Bits & Bytes
2800 Straight Rd., Fredonia
Computer Corner
200 Hamilton Ave., White Plains
Computer Era Corp.
1570 3rd Ave., New York
Computer Factory
485 Lexington Ave., NYC
Computer House, Inc.
721 Atlantic Ave., Rochester
Computerland of Nassau
79 Westbury Ave., Carle Place
Computerland of New York City
58 W. 44th St., New York
Computer World
519 Boston Post Rd., Port Chester
Comtek Electronics, Inc.
2666 Coney Island Ave., Brooklyn
Comtek Electronics, Inc.
Staten Island Mall
Store 220A, Staten Island
Digibyte Systems Corp.
31E. 31st St., New York
80-Microcomputer Services
118 Masten Ave., Cohoes
Home Computer Center
671 Monroe Ave., Rochester
Mr. Computer
Imp. Plaza, Rte. 9, Wappingers Falls
Softron Systems
308 Columbia Turnpike, Rensselaer
The Computer Tree Inc.
409 Hooper Rd., Endwell
Upstate Computer Shop
629 French Rd., Campus Plaza
New Hartford

North Carolina
Byte Shop of Raleigh
1213 Hillsborough St., Raleigh
Sound Mill
Slocum Shopping Ctr., Havelock

Ohio
Altair Business Systems, Inc.
5252 North Dixie Dr., Dayton
Astro Video Electronics
504 E. Main St., Lancaster
Cincinnati Computer Store
4816 Interstate Dr., Cincinnati
Computerland
4579 Great Northern Blvd., N. Olmstead
Computerland
6429 Busch Blvd., Columbus
Computerland
1288 Som Rd., Mayfield Heights
Computer Store of Toledo
18 Hillwyck Dr., Toledo
Forbees Microsystems Inc.
35 N. Broad, Fairborn
Microcomputer Center
7900 Paragon Rd., Dayton
Micro-Mini Computer World
74 Robinwood, Columbus
21st Century Shop
16 Convention Way, Cincinnati
Universal Amateur Radio, Inc.
1280 Aida Dr., Columbus

Oklahoma
Sounds, Etc.
Hyw. 33, Watonga
Vern Street Products
Radio Shack Dealer
114 W. Taft St., Sapulpa

Oregon
Computerland of Portland
12020 S.W. Main St., Tigard
Computer Pathways Unlimited, Inc.
2151 Davcor St. S.E., Salem
TRS-80 Products Ltd.
3520 S.E. Vineyard Rd., Portland

Pennsylvania
Artco Elect.
302 Wyoming Ave., Kingston
Artco Etc.
Back Mountain Shop. Ctr. Shavertown
Audio Mart
518 Fifth Ave., New Brighton
Computer Workshoppe
3848 William Penn Hwy, Monroeville
Computerland of Harrisburg
4644 Carlisle Pike, Mechanicsburg
Computerland of Pittsburgh
5499 William Flynn Hwy., Gibsonia
Erie Computer Co.
2127 West 8th St., Erie
Mighty Byte Computer Center
537 Easton Rd., Horsham
Personal Computer Corp.
24-26 West Lancaster Ave., Paoli
Personal Computer Corp.
Frazer Mall, Lancaster Ave., Frazer

Rhode Island
Computer City
165 Angell St., Providence
Digital World, Inc.
329 Bald Hill Rd., Warwick

South Dakota
CB Radio Shack
21st and Broadway, Yankton

Tennessee
ACS
1100 8th Ave. So., Nashville
Computerlab
671 S. Menden Hall Rd., Memphis
H & H Electronics Inc.
509 N. Jackson St., Tullahoma

Texas
Computer Port
2142 N. Collins, Arlington
Houston Computer Tech
5313 Bissonet, Bellarie
Interactive Computer
7620 Dashwood, Houston
K.A. Elect.
9090 Stemmons Frwy., Dallas
Pan American Elect. Inc.
1117 Conway, Mission
Radio Shack Dealer
21969 Katy Freeway, Katy
The Compute Shop
6353 Camp Bowie Blvd., Ft. Worth
Waghalter Books Inc.
3 Greenway Plaza E., Houston

Utah
DC Computer Co.
1911 West 70 South, Provo
Quality Technology
470 E. 2nd So., Salt Lake City

Virginia
Computer Works
Rte. 6, Box 65A, Harrisonburg
Home Computer Center
2927 Virginia Beach Blvd. Virginia Beach
Southside Radio Comm.
135 Pickwick Ave., Colonial Heights

Washington
American Mercantile Co. Inc.
2418 1st Ave. S., Seattle
Byte Shop of Bellevue
14701 N.E. 20th St., Bellevue
Computer Connection Inc.
3100 NW Bucklin Hill Rd., Silverdale
Computerland of South King Co.
1500 S. 336 St., Suite 12
Federal Way
Personal Computers
S 104 Freva, Spokane
Ye Old Computer Shop
1301 G. Washington, Richland

West Virginia
The Computer Corner Inc.
22 Beechurst Ave., Morgantown
The Computer Store
Municipal Parking Bldg., Charleston

Wisconsin
Byte Shop Of Milwaukee
6019 West Layton Ave., Greenfield
Computerland
690 S. Whitney Way, Madison
Computerworld
3015 W. Wisconsin Ave., Appleton
Petted Microsystems
4265 W. Loomis Rd., Milwaukee

Wyoming
Computer Concepts
1104 Logan Ave., Cheyenne

Puerto Rico
The Microcomputer Store
1568 Ave. Jesus T. Pinero
Caparra Terrace

Canada
CANADIAN DISTRIBUTORS:
Micron Distributing
409 Queen St., W. Toronto, Ont.
M5V 2A5

Computerland of Winnipeg
715 Portage Ave., Winnipeg, Man.
Compumart
411 Roosevelt Ave., Ottawa, Ontario
Micromatic Systems Inc.
101 8136 Park Rd., Richmond, B.C.
Micro Shack of W. Canada
333 Park Street, Regina, Sask.
Orthon Holdings Ltd.
12411 Stony Plain Road
Edmonton, Alberta
Total Computer Systems
Ajax, Ontario

Instant Software™

PETERBOROUGH, N.H. 03458
603-924-7296

Q & A on Printers and Terminals

Part 3 in a series.

David Price
3901 Victoria Lane
Midlothian, VA 23113

Compared to the high-powered technology that goes into the rest of a microcomputer, input/output devices aren't that interesting. Although terminals and printers may be among the more mundane aspects of a microcomputer, they are also among the more necessary and, in some cases, the most costly. On the other hand, declining prices make it possible for the careful shopper to find good bargains in top-flight equipment. First, however, you have to know what you're looking for.

Video terminals

Q: Having both a video terminal and a printer seems redundant. Why not use a combined keyboard-printer (such as a Teletype) and forget the video display?

A: Some people do it that way. In fact, the Teletype model 33 is one of the best selling terminals ever.

Video terminals, however, have many advantages over printing terminals of comparable price. First, video terminals are silent. Second, they do not consume paper or ribbon. Third, they are faster. Fourth, the better ones can do razzle-dazzle special effects that a printer can't do.

Q: What sort of special effects?

A: The most common is *cursor control.* A cursor is a special character—typically a small rectangle—that shows where the *next* character of text is displayed. As a new character appears on the screen, the cursor automatically advances to the next position. Cursor control allows the programmer to make the cursor move elsewhere. If the cursor was in the middle of line 15, for example, the computer might issue a command instructing the video terminal to relocate the cursor to the beginning of line 12. All ensuing transmissions would be displayed starting at line 12 instead of the old location.

Several effects can make an important word or message stand out on the screen. A word displayed in *reverse video* consists of black characters against a white background, instead of the usual white-on-black. *Blinking* characters, as the name suggests, are those that the host computer has specified to blink on and off. *Half intensity* displays some characters brighter than others.

Q: How many characters of text can a video display hold at one time?

A: That depends on the page format of the terminal. Two common formats are 64 X 16 (64 characters by 16 lines) and 80 X 24. Terminals that use low-bandwidth monitors or home TV sets are often restricted to formats of lower density, such as 32 X 16. At the other extreme, some monitors, such as the Motorola M4408, can handle 132 characters on 48 rows with no sacrifice in clarity.

Q: How are characters formed on the screen?

A: The usual method is to form characters on a *dot matrix*, which is a rectangular grid of a predetermined size, such as 5 X 7 or 7 X 9. Placing dots on all the points of the imaginary grid yields a solid rectangle. By selectively printing dots only on certain points, the terminal can create dot configurations that resemble printed characters such as letters, numbers, and punctuation.

The more points on the grid, the better some characters will look. A 7 X 9 dot matrix, therefore, is preferable to a 5 X 7 matrix. Terminals using smaller matrices are also limited in other respects. They often cannot produce lowercase letters, for

Photo 1. Two dot-matrix printers. The printer in the foreground is KSR (keyboard send receive); the one in the background is RO (receive only). (Courtesy Centronics Corp.)

Photo 2. A friction-feed printer. Paper is controlled with platen friction instead of a pin tractor. (Courtesy Centronics Corp.)

example. Some of them try to generate lowercase, but the letters are hard to read because the 5 X 7 matrix allows no room for descenders, such as the tail of a *p* or a *g*.

Many printers use dot matrix too. In fact, one recently introduced printer has an *n* X 9 matrix—that is, it allows variable spacing between characters.

Q: Can video terminals draw pictures?

A: Some can. It can be a very useful feature, too. Integrated "appliance" computers like the TRS-80 and the Apple almost always have provisions for screen graphics.

From the programmer's viewpoint, the screen of a graphics terminal is a great big dot matrix. Instead of having dimensions of 5 X 7 or 7 X 9, though, they might be 280 X 192 (Apple II Hi-Res) or 128 X 128 (Compucolor II). To plot graphs on the screen, the computer specifies the coordinates of the screen locations desired.

A few terminals can display graphics in color. An early product of this type for microcomputers was the TV Dazzler, which used a 64 X 64 grid. At each point of the grid, you could select between half and full intensity and specify any combination of the three primary colors.

Q: Is it possible to create graphics without making the program calculate every point?

A: A couple of shortcuts let a user "draw" the points himself. The first is a *digitizer*, an auxiliary device that allows a user to trace over existing drawings. A digitizer consists of a working surface (the *tablet*), a moveable crosshairs viewer and controlling electronics. You could lay a map on the tablet and trace over the desired route with the viewer. The unit would sense the position of the viewer as it moved about the map and transmit the resulting coordinates to the computer.

The second device is a *light pen*. With a light pen, you can point to a location on the terminal display screen. The terminal determines the coordinates of that location. Some of the new single-chip video display controllers, such as the Intel 8275, have provisions for light-pen input.

Printers

Q: Why does output from a printer look so peculiar?

A: You are referring to a specific type of printer that uses dot-matrix characters. Just as with a video terminal, dot matrix characters from a printer look odd.

Q: How does the printer put the dots on paper?

A: Several methods are used.

One is direct impact, where a column of wires move across the paper, punching the dots through a fabric ink ribbon. Impact printing has the advantage of being able to handle multipart carbon forms; the force of the wires can affect carbon paper the same way an ordinary typewriter does.

A second method is thermal printing. This approach also uses moving wires; instead of striking a ribbon, however, they use heat to form impressions on special heat-sensitive paper.

A third method is inkjet printing. Inkjet uses no print wires. Instead, a column of ink ejectors squirts tiny droplets of ink on the paper. Unlike thermal printing, impact and inkjet can use ordinary, untreated paper.

Q: What other types of printers are there besides these dot-matrix designs?

A: Most of the others form whole characters at a time, as a typewriter does.

Cylinder printers (e.g., Teletype 33) and band printers provide dump-grade output, while daisy-wheel and type-ball printers provide letter-grade output. The type cylinder of a cylinder printer has raised characters positioned around its circumference. The characters are in regular rows and columns; when a character is called for, the cylinder moves vertically to the desired row and spins to the desired column. A hammer behind the cylinder then hits the cylinder

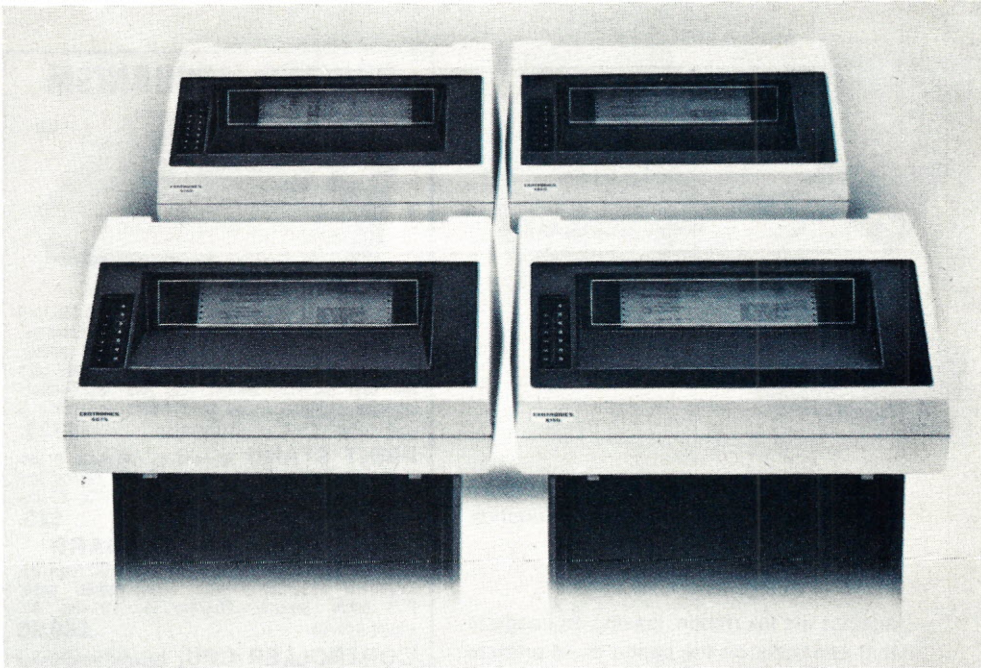


Photo 3. A quartet of band printers. (Courtesy Centronics Corp.)

Photo 4. A video terminal with detached keyboard. (Courtesy Soroc Technology)

Photo 5. A speech-recognition subsystem for microcomputers. (Courtesy Heuristics, Inc.)

against the ink ribbon, leaving an image of that character on the paper. Band printers are similar, except they use type bands, which are continuous loops with raised characters at regular intervals on the outside of the loop.

Daisy wheels and their new variant, the print thimble, provide output comparable to a good office typewriter's. Type-ball printers—the popular Selectric mechanism is an example—also provide high-quality output. The printheads of these units are usually designed for easy removal, so the user can select from printheads containing different character styles.

Q: What about those big-technology laser printers? What about those page printers that spit out a zillion lines a minute? What about . . .

A: Forget them. At least for now, they're bigger than the checkbook of even the most dedicated, self-indulging hobbyist.

Q: Well, how much do they cost?

A: It's like this: if you have to ask, you can't afford one. ■

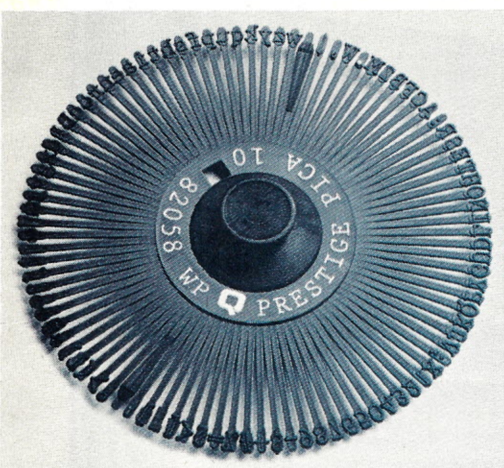


Photo 6. A daisy wheel. These are used as the type elements in many letter-quality printers. (Courtesy Qume Corp.)

Integer Choice Game for Compucolor

The computer is sometimes easy to beat, but don't be deceived.

David B. Suits
Rochester Institute of Technology
Rochester, NY 14623

This game program does not rely on an elaborate (or even a simple) statistical analysis of the player's past moves. In deciding its moves, the computer knows nothing—and cares nothing—of the player's moves. Consequently, the computer will sometimes be easy to beat—if you understand the program.

But for players who have not analyzed the program, the game can be a source of frustration. From the player's point of view, it often seems as though the computer is engaged in a remarkably deceptive strategy of second-guessing the player's moves, and the computer can end up the winner by a significant margin.

For references to discussions of the strategy this program makes use of, see Martin Gardner's column in the April 1975 issue of *Scientific American*.

Lines 430, 500 and 860 merely slow down the output so that data doesn't flash by too quickly on the display. You can omit these lines for hard-copy output or slower displays.

I wrote this program in Compucolor Disk BASIC 8001 V6.78 on my Compucolor II; the program requires less than 3K bytes. Compucolor Disk BASIC allows variable names of any length (but looks at only the first two characters). I have taken advantage of this feature by using names that are descriptive of their functions. The program should be easy to translate into any Microsoft BASIC with only minor changes. Specifically, the PLOT statements are color commands, so omit them if you are working in black and white. ■

Program listing. Integer Choice Game program in Compucolor Disk BASIC.

```
10 PLOT 6,6,12,14
20 PRINT TAB( 20 )"THE INTEGER CHOICE GAME"
30 PLOT 15
40 PRINT
50 REM      FROM MARTIN GARDNER'S
60 REM      'MATHEMATICAL GAMES',
70 REM      SCIENTIFIC AMERICAN, MARCH & APRIL, 1975
80 REM
90 REM      WRITTEN IN COMPUCOLOR DISK BASIC 8001 V6.78
100 REM      BY D.B.SUITS
110 REM     OCTOBER,11 A.L.
120 REM
130 PLOT 18:PRINT "THINK OF A POSITIVE INTEGER."
140 PRINT "I'LL THINK OF ONE, TOO."
150 PRINT "(OR THREE, OR FOUR....  HA! HA!)"
160 PRINT
170 PRINT "THEN WE'LL COMPARE OUR NUMBERS."
180 PRINT
190 PRINT "WHOEVER HAS PICKED THE SMALLER NUMBER WINS A POINT,"
200 PRINT "UNLESS IT IS SMALLER BY ONLY 1, IN WHICH CASE THE"
210 PRINT "OTHER GUY GETS 2 POINTS."
220 PRINT
230 PRINT "FOR EXAMPLE, IF YOUR NUMBER IS 15 AND MINE IS 20,"
240 PRINT "THEN YOU WIN A POINT."
250 PRINT "BUT IF YOU PICK 15 AND I PICK 16, THEN I WIN 2 "
260 PRINT "POINTS--YOU GET NONE."
270 PRINT
280 PRINT "SIMPLE, ISN'T IT?"
290 PRINT
300 PRINT
310 USER= 0:REM      USER'S SCORE
320 ME= 0:REM        MY SCORE
330 TIME= 0:REM      EVERY 5 PLAYS THE SCORE WILL BE PRINTED
340 REM
350 REM    **** GET THE NUMBERS ****
360 REM
370 MYNUM= 1:REM     MY NUMBER
380 RANNUM= RND (1)
390 IF RANNUM> .0625THEN MYNUM= 2
400 IF RANNUM> 0.375THEN MYNUM= 3
410 IF RANNUM> 0.625THEN MYNUM= 4
420 IF RANNUM> 0.9375THEN MYNUM= 5
430 FOR J= 1TO 500:NEXT J:REM   SLOW DOWN A BIT
440 PRINT :PLOT 22
450 PRINT "I HAVE MY NUMBER."
460 PRINT "WHAT IS YOUR NUMBER? ";:PLOT 17:INPUT "";PLAYERNUM
470 IF PLAYERNUM> 0AND PLAYERNUM= INT (PLAYERNUM)THEN 520
480 PLOT 21:PRINT "   OH, A WISE GUY, EH?"
490 PRINT "   OR CAN'T YOU READ INSTRUCTIONS?"
500 PRINT "   POSITIVE INTEGERS ONLY, PLEASE."
510 GOTO 440
520 PLOT 18:PRINT TAB( 7 )"MY NUMBER IS:";:PLOT 17:PRINT MYNUM
530 PRINT
```

Sample run.

```
LOAD"INTEGE":RUN
                    THE INTEGER CHOICE GAME

THINK OF A POSITIVE INTEGER.
I'LL THINK OF ONE, TOO.
(OR THREE, OR FOUR....  HA! HA!)

THEN WE'LL COMPARE OUR NUMBERS.

WHOEVER HAS PICKED THE SMALLER NUMBER WINS A POINT,
UNLESS IT IS SMALLER BY ONLY 1, IN WHICH CASE THE
OTHER GUY GETS 2 POINTS.

FOR EXAMPLE, IF YOUR NUMBER IS 15 AND MINE IS 20,
THEN YOU WIN A POINT.
BUT IF YOU PICK 15 AND I PICK 16, THEN I WIN 2
POINTS--YOU GET NONE.

SIMPLE, ISN'T IT?
```

```
540 REM    ****  FIND THE WINNER  ****
550 REM
560 FOR J= 1TO 500:NEXT J:REM  SLOW DOWN
570 PLOT 19
580 IF MYNUM- PLAYERNUM= 0THEN 770
590 IF MYNUM- PLAYERNUM< - 1THEN 740
600 IF MYNUM- PLAYERNUM= - 1THEN 710
610 IF MYNUM- PLAYERNUM> 1THEN 680
620 REM
630 REM    ****  PRINT RESULTS  ****
640 REM
650 PRINT "GOTCHA!  2 POINTS FOR ME."
660 ME= ME+ 2
670 GOTO 800
680 PRINT "1 POINT FOR YOU."
690 USER= USER+ 1
700 GOTO 800
710 PRINT "YOU WIN 2 POINTS ON THAT ONE."
720 USER= USER+ 2
730 GOTO 800
740 PRINT "1 POINT FOR ME."
750 ME= ME+ 1
760 GOTO 800
770 PRINT "TIE!  WE'LL HAVE TO TRY AGAIN."
780 PRINT
790 GOTO 350
800 TIME= TIME+ 1
810 IF TIME< 5THEN 350
820 REM
830 REM    ****  GIVE SCORE  ****
840 REM
850 PRINT
860 FOR J= 1TO 500:NEXT J:REM   SLOW DOWN
870 PLOT 23:PRINT "THE SCORE IS NOW:"
880 PLOT 21:PRINT "------------------"
890 PLOT 23:PRINT "ME:"ME;TAB( 10)"YOU:"USER
900 PLOT 21:PRINT "------------------"
910 PRINT
920 TIME= 0
930 PLOT 17:INPUT "DO YOU WISH TO CONTINUE? ";ANSWER$
940 IF LEFT$ (ANSWER$,1)= "Y"THEN 350
950 PLOT 19:PRINT
960 IF ME> USERTHEN 1020
970 IF ME< USERTHEN 1000
980 PRINT "SORRY TO SEE YOU QUIT WHEN WE'RE TIED."
990 GOTO 1040
1000 PRINT "QUITTING WHILE YOU'RE AHEAD, I SEE!  SMART!"
1010 GOTO 1040
1020 PRINT "I DON'T BLAME YOU."
1030 PRINT "BETTER LUCK NEXT TIME."
1040 PRINT
1050 PRINT
1060 END
READY
```

```
I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 1
       MY NUMBER IS: 3

1 POINT FOR YOU.

I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 3
       MY NUMBER IS: 1

1 POINT FOR ME.

I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 4
       MY NUMBER IS: 3

YOU WIN 2 POINTS ON THAT ONE.

I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 5
       MY NUMBER IS: 3

1 POINT FOR ME.

I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 1
       MY NUMBER IS: 4

1 POINT FOR YOU.

THE SCORE IS NOW:
------------------
ME: 2     YOU: 4
------------------

DO YOU WISH TO CONTINUE? Y

I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 2
       MY NUMBER IS: 2

TIE!  WE'LL HAVE TO TRY AGAIN.

      •
      •
      •

I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 2
       MY NUMBER IS: 4

1 POINT FOR YOU.

I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 3
       MY NUMBER IS: 3

TIE!  WE'LL HAVE TO TRY AGAIN.


I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 2
       MY NUMBER IS: 3

GOTCHA!  2 POINTS FOR ME.

I HAVE MY NUMBER.
WHAT IS YOUR NUMBER? 4
       MY NUMBER IS: 2

1 POINT FOR ME.

THE SCORE IS NOW:
------------------
ME: 7     YOU: 6
------------------

DO YOU WISH TO CONTINUE? N

I DON'T BLAME YOU.
BETTER LUCK NEXT TIME.
```

No.15:

Take a byte.

Software for most popular 8080/Z80 computer disk systems including*
***NORTH STAR, iCOM, MICROPOLIS, DYNABYTE DB8/2 & DB8/4, EXIDY SORCERER, SD SYSTEMS, ALTAIR, VECTOR MZ, MECA, 8" IBM, HEATH H17 & H89, HELIOS, IMSAI VDP42 & 44, REX, NYLAC, INTERTEC SUPER-BRAIN, VISTA V80 and V200, TRS-80* MODEL I and MODEL II, ALTOS, OHIO SCIENTIFIC, DIGI-LOG, KONTRON PSI-80, IMS 5000** diskette formats and **CSSN BACKUP** cartridge tapes.*

Prices reflect distribution on 8" single density diskettes. If a format is requested which requires additional diskettes, a surcharge of $8. per additional diskette will be added. A surcharge of $25 will be added for software on CSSN format DC 300XL cartridges. Media charge for 5440 disk is $100.

CP/M® VERSION 2 FOR TRS-80 MODEL II NOW AVAILABLE Genuine CP/M for Apple II coming soon! Call for details. **All Lifeboat programs require CP/M, unless otherwise stated.**

Software with Manual / Manual Alone

☐ **CP/M* FLOPPY DISK OPERATING SYSTEM** — Digital Research's operating system configured for many popular micro-computers and disk systems:

System	Version	Price
North Star Single Density	1.4	145/25 v
North Star Double Density	1.4	145/25
North Star Double/Quad	2.x	170/25
Durango F-85	2.x	170/25
iCOM Micro-Disk 2411	1.4	145/25
iCOM 3712	1.4	170/25 v*
iCOM 3812	1.4	170/25 *
Mits 3202/Altair 8800	1.4	145/25
Heath H8 + H17	1.4	145/25 m
Heath H89	1.4	145/25 m
Heath H89 by Magnolia	1.4	250/25◊
Heath H89 by Magnolia	2.x	300/25◊
Onyx C8001	2.x	300/25◊
TRS-80 Model I	1.4	145/25 m
TRS-80 Model II	2.x	170/25
TRS-80 Model II + Corvus	2.x	250/25
Processor Technology Helios II	1.4	145/25
Cromemco System 3	1.4	145/25
Intel MDS Single Density	1.4	145/25
Intel MDS Single Density	2.x	170/25
Micropolis Mod I	1.4	145/25 v
Micropolis Mod II	1.4	145/25 v

The following configurations are scheduled for release soon:

System	Version	Price
Apple II SoftCard	2.x	350/25◊
North Star Double/Quad + Corvus	2.x	250/25
North Star Horizon HD-1	2.x	250/25
Ohio Scientific C3	2.x	200/25
Ohio Scientific C3-C	2.x	250/25
Micropolis Mod II	2.x	200/25
Mostek MDX STD Bus System	2.x	350/25 **
iCOM 3812	2.x	225/25 *
iCOM 4511/Pertec D3000	2.x	375/25 * +

*Software consists of the operating system, text editor, assembler, debugger and other utilities for file management and system maintenance. Complete set of Digital Research's documentation and additional implementation notes included. Systems marked * and ** include firmware on 2708 and 2716. Systems marked + include 5440 media charge. Systems marked Ⓜ require the special Ⓜ versions of software in this catalog. Systems marked v have minor variants available to suit console interface of system. Call or write for full list of options. ◊ Includes hardware addition to allow our standard versions of software to run under it.*

DIGITAL RESEARCH

☐ **MP/M** — Installed for single density MDS-800. Multi-processing derivative of the CP/M operating system. Manual includes CP/M2 documentation ....$300/$50

☐ **MAC-8080** — Macro assembler. Full Intel macro definitions. Pseudo Ops include RPC, IRP, REPT, TITLE, PAGE, and MACLIB. Produces absolute hex output plus symbol table file for use by SID and ZSID (see below) ....$120/$15

☐ **SID-8080** — Symbolic debugger. Full trace, pass count and breakpoint program testing. Has back-trace and histogram utilities. When used with MAC, provides full symbolic display of memory labels and equated values ....$105/$15

☐ **ZSID-Z80** — Symbolic debugger with all features of SID ....$130/$15

Back Again ☐ **TEX** — Text output formatter to create paginated, page-numbered and justified copy. Output can be directed to printer or disk ....$105/$15

☐ **DESPOOL** — Utility program to permit simultaneous printing from text files while executing other programs ....$80/$10

☐ **Z80 DEVELOPMENT PACKAGE** — Consists of: (1) disk file line editor, with global inter and intra-line facilities; (2) Z80 relocating assembler, Zilog/Mostek mnemonics, conditional assembly and cross reference table capabilities; (3) linking loader producing absolute Intel hex disk file ....$95/$20

☐ **ZDT** — Z80 Monitor Debugger to break and examine registers with standard Zilog/Mostek mnemonic disassembly displays. $35 when ordered with Z80 Development Package ....$50/$10

AVOCET SYSTEMS

☐ **XASM-68** — Non-macro cross-assembler with nested conditionals and full range of pseudo operations. Assembles from standard Motorola MC6800 mnemonics to Intel hex ....$200/$25

☐ **XASM-65** — As XASM-68 for MOS Technology MCS-6500 series mnemonics ....$200/$25

New ☐ **XASM-48** — As XASM-68 for Intel MCS-48 and UPI-41 families ....$200/$25

New ☐ **XASM-18** — As XASM-68 for RCA 1802 ....$200/$25

☐ **DISTEL** — Disk based disassembler to Intel 8080 or TDL/Xitan Z80 source code, listing and cross reference files, Intel or TDL/Xitan pseudo ops optional. Runs on 8080 ....$65/$10

☐ **DISILOG** — As DISTEL to Zilog/Mostek mnemonic files ....$65/$10

☐ **SMAL/80** Structured Macro Assembler Language — Package of powerful general purpose text macro processor and SMAL structured language compiler. SMAL is an assembler language with IF-THEN-ELSE, LOOP-REPEAT-WHILE, DO-END, BEGIN-END constructs ....$75/$15

☐ **tiny C** — Interactive interpretive system for teaching structured programming techniques. Manual includes full source listings ....$105/$50

New Features ☐ **BDS C COMPILER** — Supports most features of language, including Structures, Arrays, Pointers, recursive function evaluation, overlays. Includes linking loader, library manager, and library containing general purpose, file I/O, and floating point functions. Lacks initializers, statics, floats and longs. Documentation includes "The C PROGRAMMING LANGUAGE" by Kernighan and Ritchie ....$145/$25

☐ **WHITESMITHS C COMPILER** — The ultimate in systems software tools. Produces faster code than a pseudo-code Pascal with more extensive facilities. Conforms to the full UNIX* Version 7 C language, described by Kernighan and Ritchie, and makes available over 75 functions for performing I/O, string manipulation and storage allocation. Linkable to Microsoft REL files. Requires 60K CP/M ....$630/$30

MICROSOFT

☐ **BASIC-80** — Disk Extended BASIC, ANSI compatible with long variable names, WHILE/WEND, chaining, variable length file records ....$325/$25

☐ **BASIC COMPILER** — Language compatible with BASIC-80 and 3-10 times faster execution. Produces standard Microsoft relocatable binary output. Includes MACRO-80. Also linkable to FORTRAN-80 or COBOL-80 code modules ....$350/$25

☐ **FORTRAN-80** — ANSI 66 (except for COMPLEX) plus many extensions. Includes relocatable object compiler, linking loader, library with manager. Also includes MACRO-80 (see below) ....$425/$25

☐ **COBOL-80** — Level 1 ANSI '74 standard COBOL plus most of Level 2. Full sequential, relative, and indexed file support with variable file names. STRING, UNSTRING, COMPUTE, VARYING/UNTIL, EXTEND, CALL, COPY, SEARCH, 3-dimensional arrays, compound and abbreviated conditions, nested IF. Powerful interactive screen-handling extensions. Includes compatible assembler, linking loader, and relocatable library manager as described under MACRO-80 ....$700/$25

☐ **MACRO-80** — 8080/Z80 Macro Assembler. Intel and Zilog mnemonics supported. Relocatable linkable output. Loader, Library Manager and Cross Reference List utilities included ....$149/$15

☐ **XMACRO-86** — 8086 cross assembler. All Macro and utility features of MACRO-80 package. Mnemonics slightly modified from Intel ASM86. Compatibility data sheet available ....$275/$25

☐ **EDIT-80** — Very fast random access text editor for text with or without line numbers. Global and intra-line commands supported. File compare utility included. ....$89/$15

New Features ☐ **PASCAL/M*** — Compiles enhanced Standard Pascal to compressed efficient Pcode. Totally CP/M compatible. Random access files. Both 16 and 32-bit integers. Runtime error recovery. Convenient STRINGs. OTHERWISE clause on CASE. Comprehensive manual (90 pp, indexed). SEGMENT provides overlay structure. INPORT, OUTPORT and untyped files for arbitrary I/O. Requires 56K CP/M. Specify 1) 8080 CP/M, 2) Z80 CP/M, or 3) Cromemco CDOS . $175/$20

☐ **PASCAL/Z** — Z80 native code PASCAL compiler. Produces optimized, ROMable re-entrant code. All interfacing to CP/M is through the support library. The package includes compiler, relocating assembler and linker, and source for all library modules. Variant records, strings and direct I/O are supported. Requires 56K CP/M ....$395/$25

New Version New Features ☐ **PASCAL/MT** — Subset of standard PASCAL. Generates ROMable 8080 machine code. Symbolic debugger included. Supports interrupt procedures, CP/M file I/O and assembly language interface. Real variables can be BCD, software floating point, or AMD 9511 hardware floating point. Includes strings enumerations and record data types. Manual explains BASIC to PASCAL conversion. Requires 32K. ....$250/$30

☐ **ALGOL-60** — Powerful block-structured language compiler featuring economical run-time dynamic allocation of memory. Very compact (24K total RAM) system implementing almost all Algol 60 report features plus many powerful extensions including string handling direct disk address I/O etc. ....$199/$20

☐ **CBASIC-2** Disk Extended BASIC — Non-interactive BASIC with pseudo-code compiler and run-time interpreter. Supports full file control, chaining, integer and extended precision variables, etc. ....$120/$15

MICRO FOCUS

☐ **STANDARD CIS COBOL** — ANSI '74 COBOL standard compiler fully validated by U.S. Navy tests to ANSI level 1. Supports many features to level 2 including dynamic loading of COBOL modules and a full ISAM file facility. Also, program segmentation, interactive debug and powerful interactive extensions to support protected and unprotected CRT screen formatting from COBOL programs used with any dumb terminal ....$850/$50

☐ **FORMS 2** — CRT screen editor. Output is COBOL data descriptions for copying into CIS COBOL programs. Automatically creates a query and update program of indexed files using CRT protected and unprotected screen formats. No programming experience needed. Output program directly compiled by STANDARD CIS COBOL ....$200/$20

Tasty lower prices!

EIDOS SYSTEMS

☐ **KISS** — Keyed Index Sequential Search. Offers complete Multi-Keyed Index Sequential and Direct Access file management. Includes built-in utility functions for 16 or 32 bit arithmetic, string/integer conversion and string compare. Delivered as a relocatable linkable module in Microsoft format for use with FORTRAN-80 or COBOL-80, etc. ....$335/$23

☐ **KBASIC** — Microsoft Disk Extended BASIC version 4.51 integrated by implementation of nine additional commands in language. Package includes KISS.REL as described above, and a sample mail list program. ....$585/$45
To licensed users of Microsoft BASIC-80 (MBASIC) ....$435/$45

☐ **XYBASIC** Interactive Process Control BASIC — Full disk BASIC features plus unique commands to handle byte rotate and shift and to test and set bits. Available in several versions:

Integer ROMable	$350/$25
Integer ROM squared	$350/$25
Integer CP/M	$350/$25
Extended ROMable	$450/$25
Extended ROM squared	$450/$25
Extended CP/M	$450/$25
Extended Disk CP/M	$550/$25
Integer CP/M Run Time Compiler	$350/$25
Extended CP/M Run Time Compiler	$450/$25

New ☐ **RECLAIM** — A utility to validate media under CP/M. Program tests a diskette or hard disk surface for errors, reserving the imperfections in invisible files, and permitting continued usage of the remainder. Essential for any hard disk. Requires CP/M version 2. ....$80/$5

☐ **BASIC UTILITY DISK** — Consists of: (1) CRUNCH-14 — Compacting utility to reduce the size and increase the speed of programs in Microsoft BASIC 4.51, BASIC-80 and TRS-80 BASIC. (2) DPFUN — Double precision subroutines for computing nineteen transcendental functions including square root, natural log, log base 10, sine, arc sine, hyperbolic sine, hyperbolic arc sine, etc. Furnished in source on diskette and documentation ....$50/$35

☐ **STRING/80** — Character string handling plus routines for direct CP/M BDOS calls from FORTRAN and other compatible Microsoft languages. The utility library contains routines that enable programs to chain to a COM file, retrieve command line parameters, and search file directories with full wild card facilities. Supplied as linkable modules in Microsoft format. ....$95/$20

☐ **STRING/80** source code available separately **$295/NA**

☐ **THE STRING BIT** — FORTRAN character string handling. Routines to find, fill, pack, move, separate, concatenate and compare character strings. This package completely eliminates the problems associated with character string handling in FORTRAN. Supplied with source ....$65/$15

☐ **VSORT** — Versatile sort/merge system for fixed length records with fixed or variable length fields. VSORT can be used as a stand-alone package or loaded and called as a subroutine from CBASIC-2. When used as a subroutine, VSORT maximizes the use of buffer space by saving the TPA on disk and restoring it on completion of sorting. Records may be up to 255 bytes long with a maximum of 5 fields. Upper/lower case translation and numeric fields supported. ....$175/$20

☐ **CPM/374X** — Has full range of functions to create or re-name an IBM 3741 volume, display directory information and edit the data set contents. Provides full file transfer facilities between 3741 volume data sets and CP/M files ....$195/$10

☐ **BSTAM** — Utility to link one computer to another also equipped with BSTAM. Allows file transfers at full data speed (no conversion to hex), with CRC block control check for very reliable error detection and automatic retry. We use it! It's great! Full wildcard expansion to send ★.COM, etc. 9600 baud with wire. 300 baud with phone connection. **Both** ends need one. Standard and Ⓜ versions can talk to one another. ....$150/$10

☐ **WHATSIT?*** Interactive data-base system using associative tags to retrieve information by subject. Hashing and random access used for fast response. Requires CBASIC-2 ....$175/$25

☐ **SELECTOR III-C2** — Data Base Processor to create and maintain multi Key data bases. Prints formatted sorted reports with numerical summaries or mailing labels. Comes with sample applications, including Sales Activity, Inventory, Payables, Receivables, Check Register, and Client/Patient Appointments, etc. Requires CBASIC-2. Supplied in source ...$295/$20

☐ **GLECTOR** — General Ledger option to SELECTOR III-C2. Interactive system provides for customized COA. Unique chart of transaction types insure proper double entry bookkeeping. Generates balance sheets, P&L statements and journals. Two year record allows for statement of changes in financial position report. Supplied in source. Requires SELECTOR III-C2, CBASIC-2 and 56K system ....$350/$25

Prices and specifications subject to change without notice.

☐ **CBS** — Configurable Business System is a comprehensive set of programs for defining custom data files and application systems without using a programming language such as BASIC, FORTRAN, etc. Multiple key fields for each data file are supported. Set-up program customizes system to user's CRT and printer. Provides fast and easy interactive data entry and retrieval **with transaction processing.** Report generator program does complex calculations with stored and derived data, record selection with multiple criteria, and custom formats. Sample inventory and mailing list systems included. **No support language required** ....$295/$40

MICRO DATA BASE SYSTEMS

☐ **HDBS** — Hierarchical Data Base System. CODASYL oriented with FILEs, SETs, RECORDs and ITEMs which are all user defined. ADD, DELETE, UPDATE, SEARCH, and TRAVERSE commands supported. SET ordering is sorted, FIFO, LIFO, next or prior. One to many set relationship supported. Read/write protection at the FILE level. Supports FILEs which extend over multiple floppy or hard disk devices.

☐ **MDBS** — Micro Data Base System. Full network data base with all features of HDBS plus multi-level read/write protection for FILE, SET, RECORD and ITEM. Explicit representation of one to one, one to many, many to many, and many to one SET relationships. Supports multiple owner and multiple record types within SETs. HDBS files are fully compatible.

☐ **HDBS-Z80** version ....$250/$40**

☐ **MDBS-Z80** version ....$750/$40**
8080 version available at $75 extra.

When ordering, specify one of the language interfaces listed below. Additional language interfaces available at time of purchase for $100 or $125 if purchased later.

**The single manual covering HDBS and MDBS when purchased alone comes without specific language interface manual. Manuals are available for the following Microsoft languages:

1) MBASIC 4.51, 2) BASIC-80 5.0, 3) Compiled BASIC or FORTRAN-80, 4) COBOL-80, 5) MACRO-80. ....$NA/$10

MICROPRO

☐ **SUPER-SORT I** — Sort, merge, extract utility as absolute executable program or linkable module in Microsoft format. Sorts fixed or variable records with data in binary, BCD, Packed Decimal, EBCDIC, ASCII, floating & fixed point, exponential, field justified, etc. Even variable number of fields per record! . $225/$25

☐ **SUPER-SORT II** — Above available as absolute program only ....$175/$25

☐ **SUPER-SORT III** — As II without SELECT/EXCLUDE ....$125/$25

New ☐ **DATASTAR** — Professional forms control entry and display system for key-to-disk data capture. Menu driven with built-in learning aids. Input field verification by length, mask, attribute (i.e. upper case, lower case, numeric, auto-dup, etc.). Built-in arithmetic capabilities using keyed data, constant and derived values. Visual feedback for ease of forms design. Files compatible with CP/M-MP/M supported languages. Requires 32K CP/M ....$350/$35

☐ **WORD-STAR** — Menu driven visual word processing system for use with standard terminals. Text formatting performed on screen. Facilities for text paginate, page number, justify, center and underscore. User can print one document while simultaneously editing a second. Edit facilities include global search and replace, Read/Write to other text files, block move, etc. Requires CRT terminal with addressable cursor positioning ....$445/$40

☐ **WORD-STAR Customization Notes** — For sophisticated users who do not have one of the many standard terminal or printer configurations in the distribution version of WORD-STAR ....NA/$95

☐ **WORD-MASTER** Text Editor — In one mode has superset of CP/M's ED commands including global searching and replacing, forwards and backwards in file in video mode, provides full screen editor for users with serial addressable-cursor terminal ....$145/$25

☐ **TEXTWRITER III** — Text formatter to justify and paginate letters and other documents. Special features include insertion of text during execution from other disk files or console, permitting recipe documents to be created from linked fragments on other files. Has facilities for sorted index, table of contents and footnote insertions. Ideal for contracts, manuals, etc. Now compatible with Electric Pencil* prepared files. ....$125/$20

Now-applications software for Microsoft's BASIC interpreter!

PEACHTREE SOFTWARE

☐ **GENERAL LEDGER** — Records details of all financial transactions. Generates a balance sheet and an income statement. Flexible and adaptable design for both small businesses and firms performing client writeup services. Produces reports as follows: Trial Balance, Transaction Registers, Balance Sheet, Prior Year Comparative Balance Sheet, Income Statement, Prior Year Comparative Income Statement and Department Income Statements. Interactive with other PEACHTREE accounting packages. Supplied in source code for Microsoft BASIC ....$990/$30

✔ Reader Service index—page 241

New Address New Phone

Lifeboat Associates, 1651 Third Avenue, N.Y., N.Y. 10028 **(212) 860-0300** Telex: 220501

Software with Manual / Manual Alone

☐ **ACCOUNTS PAYABLE** — Tracks current and aged payables and incorporates a check writing feature. Maintains a complete vendor file with information on purchase orders and discount terms as well as active account status. Produces reports as follows: Open Voucher Report, Accounts Payable Aging Report and Cash Requirements. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft BASIC .... $990/$30

☐ **ACCOUNTS RECEIVABLE** — Generates invoice register and complete monthly statements. Tracks current and aged receivables. Maintains customer file including credit information and account status. The current status of any customer account is instantly available. Produces reports as follows: Aged Accounts Receivable, Invoice Register, Payment and Adjustment Register and Customer Account Status Report. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft BASIC .. $990/$30

☐ **PAYROLL** — Prepares payroll for hourly, salaried and commissioned employees. Generates monthly, quarterly and annual returns. Prepares employee W-2's. Includes tables for federal withholding and FICA as well as withholding for all 50 states plus up to 20 cities from pre-computed or user generated tables. Will print checks, Payroll Register, Monthly Summary and Unemployment Tax Report. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft BASIC .... $990/$30

☐ **INVENTORY** — Maintains detailed information on each inventory item including part number, description, unit of measure, vendor and reorder data, item activity and complete information on current item costs, pricing and sales. Produces reports as follows: Physical Inventory Worksheet, Inventory Price List, Departmental Summary Report, Inventory Status Report, The Reorder Report and the Period-to-Date and Year-to-Date reports. Supplied in source code for Microsoft BASIC .... $1190/$30

GRAHAM-DORIAN SOFTWARE SYSTEMS

☐ Comprehensive accounting software written in CBASIC-2 and supplied in source code. Each software package can be used as a stand-alone system or integrated with the General Ledger for automatic posting to ledger accounts. Requires CBASIC-2.

GENERAL LEDGER	$995/$35
ACCOUNTS PAYABLE	$995/$35
ACCOUNTS RECEIVABLE	$995/$35
INVENTORY SYSTEM	$590/$35
JOB COSTING	$995/$35
APARTMENT MANAGEMENT	$590/$35
CASH REGISTER	$590/$35

☐ **POSTMASTER** — A comprehensive package for mail list maintenance that is completely menu driven. Features include keyed record extraction and label production. A form letter program is included which provides neat letters on single sheet or continuous forms. Compatible with NAD files. Requires CBASIC-2 .... $150/$15

New Features

STRUCTURED SYSTEMS GROUP

☐ **GENERAL LEDGER** — Interactive and flexible system providing proof and report outputs. Customization of COA created interactively. Multiple branch accounting centers. Extensive checking performed at data entry for proof, COA correctness, etc. Journal entries may be batched prior to posting. Closing procedure automatically backs up input files. Now includes Statement of Changes in Financial Position. Requires CBASIC-2 .... $1250/$25

☐ **ACCOUNTS RECEIVABLE** — Open item system with output for internal aged reports and customer-oriented statement and billing purposes. On-Line Enquiry permits information for Customer Service and Credit departments. Interface to General Ledger provided if both systems used. Requires CBASIC-2. .... $1250/$25

☐ **ACCOUNTS PAYABLE** — Provides aged statements of accounts by vendor with check writing for selected invoices. Can be used alone or with General Ledger and/or with NAD. Requires CBASIC-2 .... $1250/$25

☐ **PAYROLL** — Flexible payroll system handles weekly, bi-weekly, semi-monthly and monthly payroll periods. Tips, bonuses, reimbursements, advances, sick pay, vacation pay, and compensation time are all part of the payroll records. Prints government required periodic reports and will post to multiple level SSG General Ledger accounts. Requires CBASIC-2 and 54K of memory .... $1250/$25

☐ **INVENTORY CONTROL SYSTEM** — Performs control functions of adding and depleting stock items, adding new items and deleting old items. Tracks quantity of items on hand, on order and back-ordered. Optional hard copy audit trail is available. Reports include Master Item List, Stock Activity, Stock Valuation and Re-order List. Requires CBASIC-2 $1250/$25

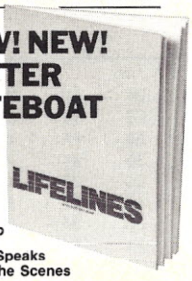
☐ **ANALYST** — Customized data entry and reporting system. User specifies up to 75 data items per record. Interactive data entry, retrieval, and update facility makes information management easy. Sophisticated report generator provides customized reports using selected records with multiple level break-points for summarization. Requires a disk sort utility such as QSORT, SUPER-SORT or VSORT and CBASIC-2. .... $250/$15

☐ **LETTERIGHT** — Program to create, edit and type letters or other documents. Has facilities to enter, display, delete and move text, with good video screen presentation. Designed to integrate with NAD for form letter mailings. Requires CBASIC-2 .. $200/$25

☐ **NAD** Name and Address selection system — Interactive mail list creation and maintenance program with output as full reports with reference data or restricted information for mail labels. Transfer system for extraction and transfer of selected records to create new files. Requires CBASIC-2 .... $100/$20

☐ **QSORT** — Fast sort/merge program for files with fixed record length, variable field length information. Up to five ascending or descending keys. Full back-up of input files created .... $100/$20

NEW! NEW! NEW! NEWSLETTER FROM LIFEBOAT



- Latest Version Numbers List of Software
- Update on CP/M Users Group
- The Great ZOSO Speaks Out from Behind the Scenes

$18 ppd. for 12 issues (U.S., Canada, Mexico). Elsewhere **$40.**
Send Check to "Lifelines," 1651 Third Avenue, New York, N.Y. 10028 or use your *VISA* or *Mastercharge*—call (212) 722-1700

★★★★★★★★ CONDIMENTS

☐ **HEAD CLEANING DISKETTE** — Cleans the drive Read/Write head in 30 seconds. Diskette absorbs loose oxide particles, fingerprints, and other foreign particles that might hinder the performance of the drive head. Lasts at least 3 months with daily use. Specify 5″ or 8″.
Single sided .... $20 each/$55 for 3
Double sided .... $25 each/$65 for 3

☐ **FLIPPY DISK KIT** — Template and instructions to modify single sided 5¼″ diskettes for use of second side in single sided drives $12.50

☐ **FLOPPY SAVER** — Protection for center holes of 5″ and 8″ floppy disks. Only 1 needed per diskette. Kit contains centering post, pressure tool and tough 7 mil mylar reinforcing rings for 25 diskettes.
5″, Kit $14.95
5″, Rings only $7.95
8″, Kit $16.95
8″, Rings only $8.95

☐ **PASCAL USER MANUAL AND REPORT** — By Jensen and Wirth. The standard textbook on the language. Recommended for use by Pascal/Z, Pascal/M and Pascal/MT users $12

☐ **THE C PROGRAMMING LANGUAGE** — By Kernighan and Ritchie. The standard textbook on the language. Recommended for use by BDS C, tiny C, and Whitesmiths C users $12

☐ **STRUCTURED MICROPROCESSOR PROGRAMMING** — By the authors of SMAL/80. Covers structured programming, the 8080/8085 instruction set and the SMAL/80 language $20

☐ **ACCOUNTS PAYABLE & ACCOUNTS RECEIVABLE—CBASIC** — By Osborne/McGraw-Hill $20

☐ **GENERAL LEDGER—CBASIC** — By Osborne/McGraw-Hill $20

☐ **LIFEBOAT DISK COPYING SERVICE** — Transfer data or programs from one media format to another at a moderate cost .... from $25

★★★★★★★★ Hearty Appetite.

*CP/M and MP/M are trademarks of Digital Research.
Z80 is a trademark of Zilog, Inc.
UNIX is a trademark of Bell Laboratories.
WHATSIT? is a trademark of Computer Headware.
Electric Pencil is a trademark of Michael Shrayer Software.
TRS-80 is a trademark of Tandy Corp.
Pascal/M is a trademark of Sorcim.

†Recommended system configuration consists of 48K CP/M, 2 full size disk drives, 24 x 80 CRT and 132 column printer.

Ⓜ Modified version available for use with CP/M as implemented on Heath and TRS-80 Model I computers.

Ⓛ User license agreement for this product must be signed and returned to Lifeboat Associates before shipment may be made.

Ⓘ Ⓧ This product Includes/eXcludes the language manual recommended in Condiments.

Ⓢ Serial number of CP/M system must be supplied with orders.
Ⓩ Requires Z80 CPU.

Ordering Information

MEDIA FORMAT ORDERING CODES
When ordering, please specify format code.

LIFEBOAT ASSOCIATES MEDIA FORMATS LIST
Diskette, cartridge disk and cartridge tape format codes to be specified when ordering software for listed computer or disk systems. All software products have specific requirements in terms of hardware or software support, such a MPU type, memory size, support operating system or language.

Computer system	Format Code
Altair 8800 Disk	See MITS 3200
Altos	A1*
Apple + Microsoft SoftCard	RG
BASF System 7100	RD
Blackhawk Single Density	Q3
Blackhawk Micropolis Mod II	Q2
CDS Versatile 3B	Q1
CDS Versatile 4	Q2
COMPAL-80	Q2
Cromemco System 3	A1*
Cromemco Z2D	R6
CSSN BACKUP (tape)	T1#
Delta	A1*
Digi-Log Microterm II	RD
Digital Microsystems	A1*
Discus	See Morrow Discus
Durango F-85	RL
Dynabyte DB8/2	R1
Dynabyte DB8/4	A1*
Exidy Sorcerer + Lifeboat CP/M	Q2
Exidy Sorcerer + Exidy CP/M	Q4
Heath H8 + H17/H27	P4
Heath 89 + Lifeboat CP/M	P4
Heath 89 + Magnolia CP/M	P7
Helios II	See Processor Technology
Horizon	See North Star
ICOM 2411 Micro Floppy	R3
ICOM 3712	A1
ICOM 3812	A1*
ICOM 4511 5440 Cartfidge CP/M 1.4	D1#
ICOM 4511 5440 Cartridge CP/M 2.2	D2#
IMS 5000	RA
IMS 8000	A1*
IMSAI VDP-40	R4**
IMSAI VDP-42	R4**
IMSAI VDP-44	R5**
IMSAI VDP-80	A1**
Intecolor	See ISC Intecolor
Intel MDS Single Density	A1
Intertec SuperBrain DOS 0.1	R7
Intertec SuperBrain DOS 0.5-2.X	RJ
Intertec SuperBrain DOS 3.X	RK
ISC Intecolor 8063/8360/8963	A1
Kontron PSI-80	RF
Meca 5¼″	P6
Micromation (Except TRS-80 below)	A1*
Micropolis Mod I	Q1
Micropolis Mod II	Q2
MITS 3200/3202	B1
Morrow Discus	A1*
Mostek	A1
MSD 5¼″	RC
North Star Single Density	P1
North Star Double/Quad	P2
Nylac Single Density	Q3
Nylac Micropolis Mod. II	Q2
Ohio Scientific C3	A3
Onyx C8001	T2#
Pertec PCC 2000	A1*
Processor Technology Helios II	B2
RAIR Single Density	R9
RAIR Double Density	RE
Research Machines 8″	A1
Research Machines 5¼″	RH
REX	Q3
SD Systems 8″	A1*
SD Systems 5¼″	R3
Sorcerer	See Exidy Sorcerer
Spacebyte	A1
SuperBrain	See Intertec
Tarbell	A1*
TEI 5¼″	R3
TEI 8″	A1*
Thinkertoys	See Morrow Discus
TRS-80 Model I 5¼″	R2
TRS-80 Model I + FEC Freedom	RN
TRS-80 Model I + Micromation	A4*
TRS-80 Model I + Omikron 5¼″	RM
TRS-80 Model I + Omikron 8″	A1
TRS-80 Model I + Shuffleboard 8″	A1
TRS-80 Model II	A1*
VDP-40/42/44/80	See IMSAI
Vector MZ	Q2
Versatile	See CDS Versatile
Vista V80 5¼″ Single Density	P5
Vista V200 5¼″ Double Density	P6
Zenith Z89 + Lifeboat CP/M	P4
Zenith Z89 + Magnolia CP/M	P7

*Single-Side Single-Density disks are supplied for use with Double-Density and Double-Side 8″ soft sector format systems.

**IMSAI formats are single density with directory offset of zero.

#A media surcharge of $25 for orders on tape formats T1 and T2 and of $100 for orders on disk formats D1 and D2 will be added.

The list of available formats is subject to change without notice. In case of uncertainty, call to confirm the format code for any particular equipment.

Prices F.O.B. New York.
Shipping, handling and C.O.D. charges extra.

Manual cost applicable against price of subsequent software purchase.

The sale of each proprietary software package conveys a license for use on one system only.

VISA master charge

ELCOMP NEW Books

Care and Feeding of the Commodore PET
Eight chapters exploring PET hardware. Includes repair and interfacing information. Programming tricks and schematics.
Order No. 150 **$11.00**

8K Microsoft BASIC Reference Manual
Authoritative reference manual for the original Microsoft 4K and 8K BASIC developed for Altair and later computers including PET, TRS-80, and OSI. OSI owners please take note!
Order No. 151 **$ 9.95**

Expansion Handbook for 6502 and 6802
(S-44 Card Manual) Describes all of the 4.5 x 6.5 44 pin S-44 cards incl. RAM, ROM, dig. I/O, MUX/A to D, EPROM Prog. etc. With schematics and funct. descriptions. A must for every KIM, SYM and AIM owner.
Order No. 152 **$ 9.95**

Microcomputer Application Notes
Reprint of Intels' most important application notes, including 2708, 8085, 8255, 8251 chips. Very necessary for the hardware buff.
Order No. 153 **$ 9.95**

Complex Sound Generation
New, revised applications manual for the Texas Instruments SN 76477 Complex Sound Generator. Circuit board available ($8.95)
Order No. 154 **$ 6.95**

Small Business Programs
Complete programs for the business user. Mailing List, Inventory, Invoice Writing and much more. Introduction into Business Applications. Many listings.
Order No. 156 **$14.90**

The First Book of Ohio Scientific, Vol. I
Contains an introduction to personal computers and describes the Ohio Scientific Line. Contains explanatory diagrams: block, hook-up, expansion, tricks, hints and many interesting listings. Hardware and software information not previously available in one compact source. 192 pages.
Order No. 157 **$ 7.95**

The First Book of Ohio Scientific, Vol. II
Vol. II contains very valuable information about Ohio Scientific microcomputer systems. Introduction to OS-65D and OS65-U, networking and distributed processing, systems specifications, business applications, hard- and software hints and tips.
Order No. 158 **$ 7.95**

Mailing List Program for Challenger C1/C2 8K
Order No. 2004 - Personal Version **$ 9.95**
Order No. 2005 - Business Version **$ 9.95**

Ohio Scientific Expansion Information
Conversion of C1P (Cassette) to 52x26 display. Detailed step-by-step instructions for doubling the C1P speed and display size!
Order No. 1105 **$12.00**

Important Software for CBM 16K/32K
Most powerful Editor/Assembler for Commodore CBM 16/32K on cassette. Very fast— Editor divides screen into 3 parts. Scrolling text window, 24 direct commands, 19 serial commands, status and error messages. Assembler can be started directly from the editor or from the TIM-monitor. Translates in three passes. If an error is encountered, automatic return to the editor. Cassette with DEMO.
Order No. 3276 **$69.00**

ATTENTION APPLE USERS
Same as above for Apple II or Apple II plus.
Order No. 3500 **$89.00**

MONJANA/1 makes Machine Language Programming easy!
In every Commodore CBM there is a spare ROM socket waiting for its MONJANA/1. The new MONJANA/1 Machine Language Monitor in ROM offers more user guidance and debugging aids than any other monitor available today. It is indispensable for anyone intending to take full advantage of the computers features. - Trace, link, disassemble, dump, relocate, line assemble and much more. Every command function has demand printout option. Price includes extensive manual.
Order No. 2001 **$98.00**

JANA-Monitor on cassette for the PET
Similar to MONJANA/1 - very powerful.
Order No. 2002 **$29.00**

ELCOMP PUBLISHING Inc. ✓116
3873-L Schaefer Ave., Chino CA 91710 (714) 591-3130

Please send me the books/software indicated below:

☐ I enclose $ ______ send postpaid
☐ Send COD ($5 extra)
☐ Charge my ☐ VISA ☐ Mastercharge

Acct. No. ______
Expir date ______ Signature ______

Book No.	Book No.	Software No.

☐ 1 Year subscription to ELCOMP Newsletter $9.80

Name ______ Phone ______
Address ______
City ______ State ______ Zip ______

CA add 6% sales tax. We also accept Euroscheck. All orders outside USA must add 15% shipping.

Memory-Checking Program for the 1802

This simple program will sniff out defective chips.

John R. Bunn
MOTU 7 PO Box 105
FPO Seattle, WA 98762

I bought a basic Elf II microcomputer from Netronics, Ltd., 333 Litchfield Rd., New Milford, CT 06776, and slowly added to it as I learned more about it. Now I have the monitor board, 8K of RAM, ASCII keyboard, Tiny BASIC and various software support. With the rf and if sections removed, a nine-inch solid-state TV serves as my monitor. The biggest problem I have with my new hobby is finding time to do things I want. Software support for the 1802 is still in its infancy, and you have to write most programs yourself.

All was going well with my Elf II until I noticed that Tiny BASIC would not work on large programs. The problem was on one of the 4K RAM boards, and I could have just swapped chips to locate the defective one. Instead, I wrote a simple memory-checking program using what I learned from working with my Elf II. The resulting 157-byte program fits in the original 256 bytes of RAM and may be used to check one byte of memory or as many bytes as you have. Fig. 1 is a flowchart of the program, and Program 1 is the machine-language listing. The program goes into memory locations 0010 to 0098, but you can put it on any page by changing the address at 0011.

After initialization, the program turns on the Q LED and waits for data input. First, the number of bytes of memory to be checked is entered, high order and then low order. Second, the starting address is entered, high address and then low address. For example, to check one page of memory starting at page 07, enter 00 FF 07 00. As the data is entered, the hex display echoes it, and once all the data is entered, the Q will go out and the program will run the memory check.

If all the memory locations can store all the test words, the Q LED flashes at a rapid rate. If a memory location fails to store a test word, the Q LED flashes at a slow rate, and the hex display shows the error address. When the Q LED is on, the high address shows on the hex display; when the Q led is off, the low address shows.

The program runs at 0010. Be careful to enter the input data correctly or disaster will most likely occur when the program runs. The memory check program found my trouble, a defective chip at location 17B3. It is a simple but effective program. ■

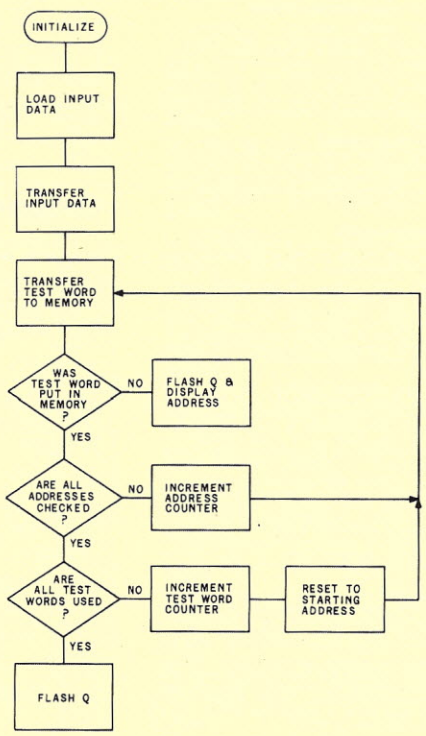


Fig. 1.

0010	F8	00	BF	BE	BD	F8	AC	AF	F8	AB	AE	F8	9A	AD	F8	04
0020	AA	F8	0D	A9	EE	F8	01	73	FE	73	FE	73	FE	73	FE	73
0030	FE	73	FE	73	FE	73	F8	AA	73	F8	00	73	F8	FF	73	F8
0040	55	73	7B	EF	37	44	3F	46	6C	64	2A	8A	3A	44	7A	F8
0050	AC	AF	EF	72	BB	72	AB	EF	72	BC	72	AC	EC	0E	5C	F3
0060	32	7A	8C	5D	1D	9C	5D	2D	ED	7A	64	F8	60	B8	28	98
0070	3A	6E	31	69	7B	64	2D	2D	30	6B	EB	2B	8B	3A	89	9B
0080	3A	89	29	89	32	8C	1E	30	4F	1C	30	5C	7A	F8	10	B8
0090	28	98	3A	90	31	8C	7B	30	8D							

Program 1.

Chaining Data With the Sorcerer

Don't let your Sorcerer torture you with limited capabilities.

Charles Dailey
3217 NE 54th St.
Vancouver, WA 98663

Filing capabilities were dismal with the 8K Microsoft BASIC that came with my Sorcerer. Exidy has a CSAVE that off-loads a program onto a cassette, but does not save the information entered into the program. A CSAVE* is available for off-loading numerical arrays. It is possible to convert strings to numerical data and then save it, but the CSAVE* command lacks the cyclical redundance-check system that makes the regular CSAVE so reliable. As a result, practical filing capabilities are minimal in Sorcerer with cassette.

The obvious way to save information is to enter it as DATA statements and then extract it with READ/DATA commands. But this means you must set up a closely structured sequence. In a mailing list, the name, always followed by the address, always followed by the city and zip, is required. Nothing can be skipped. If the zip is not available, you still must make a field for the one not available. This is confining and cumbersome.

Having several hundred 35 mm slides to index and cross-index, I wrote a program to identify any properly titled slide. The key to making this idea work was to input a word that would be found in the title and have the Sorcerer "read" all the data until it found it, then print that data statement. The "reading" would utilize the MID$(D$,L,I) command, where D$ is the data string, L is the loop number and I is the length of the input word or string.

The system worked well, and it was simple to add refinements along the way—a counter terminating in an input command to keep the screen from scrolling, for instance.

The next problem was to figure a way to include more than one data line in the final printout. Slide titles could be stated in one line or less than 50 characters, but not data about equipment and supplies, sources of supply, prices and other relevant information. I needed a way to come back later and add new information without retyping the entire DATA line, so I built upon the plan used for the slide catalog and added the chaining capability.

To avoid having to retype the existing DATA line, I decided to begin the lines concatenated to the previous line with a plus sign. Leaving ten program lines between subjects gives you the option of adding nine additional lines of data, or of returning on nine later occasions to update the original data line.

```
100 CLEAR 5000:PRINT:PRINT:GOSUB 400C
300 PRINT'                    EQUIPMENT CATALOG':      PRINT
500 PRINT'Type in the word to be searched for in CAPITAL LETTERS
600 PRINT
700 INPUT '          ';I$
800 I = LEN(I$):RESTORE
850 FOR L=5 TO 9:E$(L)=CHR$(32):NEXT L
900 GOSUB 4000:C=0
950 PRINT 'Checking for 'I$
1000 READ D$
1010 :IF D$='DONE'AND C<5 THENPRINT:PRINTD$:GOSUB 4050:GOTO 600
1020 :IF D$ = 'DONE' THEN GOTO 1040
1030 GOTO 1100
1040 PRINT:GOSUB 4400:PRINT D$:GOSUB 4050:GOTO 600
1100 :X=LEN(D$):IFLEFT$(D$,1)='+'THENE$=E$+RIGHT$(D$,X-1):GOTO1000

1300 :IFLEFT$(D$,1)<>'+'THENGOSUB4200:E$=CHR$(32):E$=D$:GOTO1000
4000 REM     SUBROUTINES SECTION **************************
4010 PRINT CHR$(12):PRINT:PRINT:RETURN
4020 :
4050 PRINT:PRINT 'You may try another word':RETURN
4090 :
4200 FOR L = 1 TO LEN(E$)
4210 :IF I$=MID$(E$,L,I) THEN GOSUB 4300:RETURN
4220 NEXT L:RETURN:  REM  I$ NOT FOUND SO RETURN TO 1300
4240 :
4300 C = C + 1
4320 IF C < 5 THEN PRINT:PRINT E$:RETURN
4340 E$(C) = E$
4360 IF C < 10 THEN RETURN:  REM:  RETURNS TO 1300
4380 PRINT:PRINT
4400 INPUT 'Press RETURN to clear screen and continue listing';Z
4410 GOSUB 4000:PRINT 'Checking for 'I$:PRINT
4420 FOR L=5 TO 9:PRINT E$(L):PRINT:NEXT L:C=4:RETURN
4510 :
5000 REM  ADD DATA HERE
5010 DATA THIS IS A SAMPLE OF THE FIRST LINE OF YOUR DATA
5011 DATA + DO THIS IF YOU WISH TO CONTINUE THE ENTRY
5012 DATA + THIS LINE MAY BE ADDED, TOO.
5020 DATA THE SECOND ENTRY WOULD BEGIN HERE.
5021 DATA + AND THIS COULD BE ADDED TO IT.
9990 DATA LIST BEGINS:   REM: THIS MUST REMAIN AT THE END
10000 DATA DONE
```

Program listing.

My procedure requires reading the data line and checking it for a leading + mark. If it does not have a + mark, it must be a new subject, so the contents of E$ are examined for the INPUT $ that is sought. If the $ is found, then E$ is printed; otherwise, D$ is put in E$ and a new D$ is read and examined for a +. If the + is present, it is removed and the rest of D$ is concatenated to E$. The next data line is then read into D$ and examined.

At this point the cycle repeats itself until it encounters a final DONE, then control transfers back and another INPUT word is called for. Note that a space should directly follow the + mark. No commas or colons are allowed in the data statements.

In its simplest form the program requires the user to read the data from the CRT and then press RETURN. This often leaves the operator sitting while the Sorcerer processes the data, so I added an array. Now while the user reads the data brought up on the tube, Sorcerer loads arrays so the user can instantly fill the tube with more data when he presses RETURN. This is a patience-saving addition to the basic plan.

This program overcomes Sorcerer's filing weakness and opens the way for storing and recalling endless amounts of DATA, limited only by available memory. ■

Reader Service index—page 241

A "Personable" Calendar

Lists, lists, lists! Let PET clear the clutter and confusion.

G. R. Boynton, Chairman
Dept. of Political Science
University of Iowa
Iowa City, IA 52242

I work best off lists. So when time becomes tight, as it frequently does, I start producing lists. My desk calendar serves as one list, and there are usually several pads scattered around that have lists on them. And in the office I am using this year, there is an entire "green" board that I use for making lists.

This proliferation of lists not only produces clutter, but it also eventually leads to confusion. I often cannot find the right list.

So when my PET arrived a few months ago, I knew that one of the first things it would have to do was help me consolidate my lists. The result is this program, which I call a calendar. It stays in the computer whenever I am not using the PET for some other purpose, and it keeps me informed about what I am supposed to be doing and when.

Introduction

Before I describe the program and its operation, there are two things that need to be said. First, since this is a rather long program, the only way I could approach programming was to break it up into manageable pieces and write them one at a time. So the program is built out of 12 blocks, many of which are subroutines and subroutines within subroutines. Each section of the program is short—ten to 20 lines—and each was, thus, easy to write and, I hope, easy to understand. But the best way to understand this program is not to start at the first line and read through the last line.

Second, I call this program a "personable" calendar. I have used many programs for machines ranging from university-owned IBMs and CDCs to the PET. The programs were all designed to be used by the impersonal other.

This program is designed to be used by a human being—me. Several of my biases about what interacting with a computer ought to be like are built into this program; I had to leave out a number of biases to keep the program within 7167 bytes and still have it perform its tasks. I am a person who wants to be recognized. I speak English and would like to interact with my computer more or less in English. I am not interested in interacting in the mode of Y, N, 1, 2, etc., even if it would save a little time.

As I go through the description of the program, I will show how it can be personalized for another human being—you. But if you are the kind of person who is in a hurry with computers, you probably will not like the "frills" associated with this program. If so, just cut them out. At the end of the article I will summarize some of the "personable" features in this program that I would like to see in other programs.

1. Control for hello	(lines 10-120)
2. Greetings Bob	(lines 1000-1099)
3. Greetings other	(lines 1100-1199)
4. Read data for calendar	(lines 2000-2140)
5. Route for calendar subroutines	(lines 2145-2299)
6. Today's events	(lines 2300-2330)
7. Other dates	(lines 2400-2450)
8. Unfinished items	(lines 2500-2599)
9. Change status of item	(lines 2600-2699)
10. Add item to calendar	(lines 2700-2799)
11. Write to tape	(lines 2800-2898)
12. Search by date	(lines 2900-2999)

Table 1. Program description

Program Description

The program is built out of twelve blocks of code. The list with line numbers is shown in Table 1. This list is repeated in lines 900–968 of the program listing. I fiddle with my programs, and I need to know what is where. REM statements may be good for some things, but they are not good for that. They are usually dispersed through the program, and they are not executable.

After this program is loaded it will describe itself if you type RUN 900 or if you type PROGRAM when the program asks "What's next?" This is a particularly valuable feature to build in as you are writing a program. Whenever you want to check some other part of the program, you just type RUN 900, and this directory tells you what line numbers to list.

The first three sections of the program are all designed to say hello. The program begins by asking the user to type hello. Then lines 30, 40 and 50 look for my name in the response. If I

am in a good mood I sometimes type: "Good morning Isaac; this is Bob." And if I am not in a good mood, I may be a little more tart.

If Isaac (my computer is named after Isaac Asimov, who practically invented the mobile computer) finds my name, then I am routed to lines 1010–1070, which produce a greeting from Isaac. It will not do for the program to be looking for my name on your computer. Just replace my name with your name in lines 40 and 60, and it will look for your name.

I wrote lines 1010–1070 because I got bored with looking at the same greeting every time I used the program. So line 1010 generates a number between 1 and 4. Then line 1020 uses this number to route to one of four greetings. You can change these greetings to make them consistent with the style of your computer. Then line 1025 pauses for a few seconds, and line 1030 returns you to the main routine.

The second part of the greeting can be used to amaze one's family and friends. If the operator just types hello, as instructed, he or she will learn the name of the computer; you will, no doubt, want to change the name in line 70. The operator is also asked his or her name and is then given a brief description of the program in lines 1110–1180. It does not get used very often on my computer, but it does surprise people who have not been around this type of computer.

The program asks for the date, and then routes to the calendar part of the program. The program stores the list of calendar activities on a tape, and lines 2010–2140 read this tape. Lines 2010–2040 remind the user that the tape on which the program is stored must be taken out of the cassette and the data tape put in. Since it is a reminder, any answer to the question other than "no" will let you proceed.

Lines 2050 through 2140 do the actual reading. The first piece of information read is N, which is the number of items on the tape. N is used to define J in line 2080, and J is used in line 2090 to set the size of the three arrays that are used for storing data. The arrays are set larger than the data to be read so there will be room for making additions later in the program.

The section of the program in lines 2150–2295 asks if you want to see a list of today's activities. Then, beginning in line 2190, it routes you through the search and write routines, which make up the rest of the program. Lines 2190–2255 display the options. The user responds by typing the first word for the appropriate option. If you want

Past items not completed

then you type

Past

I thought of a number of ways to assist the uninformed user about which options would not be executed for the informed user, but I finally decided that it was not worth the extra lines of program. So you just have to know to type the first word. Lines 2260 through 2285 convert the English into numbers, which are then used in line 2290 to route the program to the appropriate subroutine.

Each entry in the calendar is called an item. There are three routines for searching the calendar. There are also two routines for changing the calendar. The best introduction to these five routines is ADDITIONS.

With ADDITIONS, you are able to add items to your calendar. It asks for three pieces of information. It asks for the item's date, which includes an item number so that items for the same date can be distinguished. If there are already two activities to be done on November 1, then the date will look like

Nov 01 03

This is the third item for November 1. Then it asks what is the item to be entered. Any message to yourself no longer than two lines will do. Finally, it asks about the status: finished or unfinished?

It is very unlikely that you will be able to remember how many items there are on the calendar for a given day, so lines 2705–2730 let you look at what is already on for that day if you do not remember. Then you input DA$(N + 1) in line 2735. DA$ is the array in which dates are stored, and (N + 1) puts the new date in the right place in the array. In line 2740 you input

Personable calendar program.

```
 1 REM
 10 PRINT"PLEASE TYPE HELLO"
 20 PRINT"":INPUT G$
 30 FOR I=1TO LEN(G$)
 40 IF MID$(G$,I,LEN("BOB"))="BOB" THEN
NA$="BOB"
 50 NEXT I
 60 IF NA$="BOB" THEN GOTO 1010
 70 PRINT"MY NAME IS ISAAC. WHAT IS YO
UR NAME?"
 80 PRINT"":INPUT NA$
 90 GOSUB 1110
 100 PRINT"WHAT IS THE DATE TODAY?"
 110 PRINT"":INPUT D$
 120 GOTO 2010
 900 PRINT"THE PROGRAM IS MADE UP OF
THE FOLLOWING"
 902 PRINT"COMPONENTS:"
 904 PRINT"1.CONTROL FOR HELLO"
 906 PRINT"10-120"
 908 PRINT"2.GREETINGS BOB"
 910 PRINT"1000-1099"
 912 PRINT"3.GREETINGS OTHER"
 914 PRINT"1100-1199"
 916 PRINT"4.READ DATA FOR CALENDAR"
 918 PRINT"2000-2140"
 920 PRINT"5.ROUTE FOR CALENDAR SUBR'
S"
 922 PRINT"2145-2299"
 924 PRINT"6.TODAY'S EVENTS"
 926 PRINT"2300-2330"
 928 PRINT"TYPE 'GO' TO GET THE REST"
 930 GET A$:IF A$="" THEN 930
 936 PRINT"7.OTHER DATES"
 938 PRINT"2400-2450
 940 PRINT"8.UNFINISHED ITEMS"
 942 PRINT"2500-2599"
 944 PRINT"9.CHANGE STATUS OF ITEM"
 946 PRINT"2600-2699"
 952 PRINT"10.ADD ITEMS TO CALENDAR"
 954 PRINT"2700-2799"
 956 PRINT"11.WRITE TO TAPE"
 958 PRINT"2800-2898"
 960 PRINT"12.SEARCH BY DATE"
 962 PRINT"2900-2999"
 968 PRINT"TYPE 'GO' TO RETURN TO CALE
NDAR"
 970 GET A$:IF A$="" THEN 970
 979 RETURN
 980 IF FRE(0)>200 GOTO 987
 981 PRINT"THERE IS VERY LITTLE SPACE L
EFT IN"
 982 PRINT"MEMORY!"
 983 PRINT"DO YOU WANT TO DELETE ALL OF
 THE ITEMS  THAT ARE FINISHED?"
 984 PRINT:INPUT A$
 985 IF A$="YES" THEN E=1:GOSUB 2510
 986 GOTO 996
 987 IF CH<=WR THEN 996
 988 PRINT"THERE ARE ONE OR MORE CHANGES
 THAT HAVE NOT BEEN RECORDED."
 989 PRINT"DO YOU WANT TO WRITE THEM TO
 TAPE?":PRINT:INPUT A$
 990 IF A$="YES" THEN GOSUB 2810
 996 PRINT"GOODBYE ";NA$
 998 PRINT"GLAD I COULD HELP YOU."
 999 END
 1000 REM HELLO BOB
 1010 LET N=INT((RND(1)*10)/2.5)+1
 1020 ON N GOSUB 1040,1050,1060,1070
 1025 FOR K=1 TO 1500:NEXT K
 1030 GOTO 100
 1040 PRINT"HELLO DESIGNER":RETURN
 1050 PRINT"HI, BOB":RETURN
 1060 PRINT"BACK TO WORK, EH?":RETURN
 1070 PRINT"HOWDY, BOB":RETURN
 1100 REM HELLO OTHER
 1110 PRINT"";NA$;" I AM A CALENDAR OF
 THINGS"
 1120 PRINT"TO BE DONE."
 1130 PRINT"I HELP BOB KEEP UP WITH HIS
 WORK."
 1135 FOR K=1 TO 1500:NEXT
 1140 PRINT"HE TELLS ME WHAT HE HAS TO
DO ON EACH"
 1150 PRINT"DAY, AND I REMIND HIM ABOUT
WHAT IS ON"
 1160 PRINT"FOR TODAY, AND WHAT HE HAS N
OT FINISHED."
 1170 FOR K=1 TO 6500:NEXT
 1180 RETURN
 2000 REM CALENDAR ROUTING
 2010 PRINT"FIRST I HAVE TO READ THE
TAPE, ";NA$;"."
 2020 PRINT"IS IT MOUNTED AND REWOUND?"
 2030 PRINT"":INPUT A$
 2040 IF A$="NO"THEN PRINT"NOTHING HAPP
ENS WITHOUT READING THE TAPE":GOTO2020
 2050 L=1:D=1:C=0:F$="DCALENDAR"
 2060 OPEN L,D,C,F$
 2070 INPUT#L,N
 2080 J=N+10
 2090 DIM DA$(J),IT$(J),ST$(J)
 2100 FOR K=1 TO N
 2110 INPUT#L,DA$(K),IT$(K),ST$(K)
 2130 NEXT K
 2140 CLOSE L
 2145 REM ROUTE FOR CALENDAR
 2150 PRINT"WOULD YOU LIKE TO SEE WHAT
 IS ON FOR"
 2160 PRINT"TODAY?"
 2170 PRINT"":INPUT A$
 2180 IF A$="YES" THEN GOSUB 2310
 2190 PRINT"WHAT'S NEXT ";NA$;"?"
 2200 PRINTTAB(3)"OTHER DATES"
```

```
 2210 PRINTTAB(3)"PAST ITEMS NOT COMPLETED"
 2220 PRINTTAB(3)"STATUS UPDATE"
 2230 PRINTTAB(3)"ADDITIONS"
 2240 PRINTTAB(3)"TODAY"
 2250 PRINTTAB(3)"DONE WITH CALENDAR"
 2255 PRINT"■":INPUT A$
 2260 IF A$="OTHER" THEN RO=1
 2265 IF A$="PAST" THEN RO=2
 2270 IF A$="STATUS" THEN RO=3
 2275 IF A$="ADDITIONS" THEN RO=4
 2280 IF A$="TODAY" THEN RO=5
 2283 IF A$="PROGRAM" THEN RO=6
 2285 IF A$="DONE" THEN 980
 2290 ON RO GOSUB 2410,2510,2605,2705,23
10,900
 2295 GOTO 2190
 2300 REM TODAY
 2310 LET SE$=D$
 2320 GOSUB 2900
 2330 RETURN
 2400 REM OTHER DATE
 2410 PRINT"■WHICH DATE ARE YOU LOOKING
FOR?"
 2420 PRINT"■":INPUT DB$
 2430 LET SE$=DB$
 2440 GOSUB 2900
 2450 RETURN
 2500 REM STATUS UNFINISHED
 2510 PRINT"■■■HERE IS WHAT IS HANGING O
VER YOUR HEAD!"
 2520 FOR K=1 TO N
 2525 IF E=1 AND ST$(K)="NOT FINISHED" T
HEN NN=NN+1
 2530 IF ST$(K)="NOT FINISHED" THEN 2560
 2540 NEXT K
 2545 IF E=1 THEN 2810
 2550 RETURN
 2560 PRINT"■"DA$(K)
 2570 PRINT IT$(K)
 2580 FOR Z=1 TO 1500:NEXT Z
 2590 GOTO 2540
 2600 REM CHANGE STATUS
 2605 PRINT"■■WHAT IS THE DATE OF THE IT
EM YOU WANT"
 2610 PRINT"TO CHANGE?"
 2615 PRINT"■":INPUT DB$
 2620 PRINT"■DO YOU WANT TO LOOK AT THE
ITEMS FOR"
 2625 PRINT"THAT DATE FIRST?"
 2630 PRINT"■":INPUT A$
 2635 IF A$="YES" THEN GOSUB 2430
 2640 PRINT"■WHAT IS THE DATE AND ITEM N
UMBER TO BE  CHANGED?"
 2645 PRINT"■":INPUT DI$
 2650 PRINT"■IS THE NEW STATUS TO BE 'FI
NISHED' OR"
 2655 PRINT"'NOT FINISHED'?"
 2660 PRINT"■":INPUT ST$
 2665 FOR K=1 TO N
 2670 IF DI$=DA$(K) THEN ST$(K)=ST$
 2675 NEXT K
 2680 CH=CH+1:PRINT"■OKAY THE CHANGE IS
MADE. DO YOU WANT TO MAKE ANOTHER CHANGE?"
 2685 PRINT"■":INPUT A$
 2690 IF A$="YES" THEN 2605
 2695 RETURN
 2700 REM ADDITIONS
 2705 PRINT"■WHAT IS THE DATE OF THE NEW
 ENTRY?"
 2710 PRINT"■":INPUT DB$
 2715 PRINT"■SINCE YOU HAVE TO GIVE AN I
TEM NUMBER"
 2720 PRINT"AS WELL AS THE DATE DO YOU W
ANT TO LOOK AT THE ITEMS FOR THAT DATE?"
 2725 PRINT"■":INPUT A$
 2730 IF A$="YES" THEN GOSUB 2430
 2735 PRINT"■WHAT IS THE DATE AND ITEM N
UMBER?":INPUT DA$(N+1)
 2740 PRINT"■WHAT IS THE ITEM TO BE ENTE
RED?":INPUT IT$(N+1)
 2745 PRINT"■WHAT IS THE STATUS; FINISHE
D OR NOT      FINISHED?":INPUT ST$(N+1)
 2750 N=N+1:CH=CH+1
 2755 PRINT"■DO YOU WANT TO ADD ANOTHER
ITEM?":INPUT A$
 2760 IF A$="YES" THEN 2705
 2765 PRINT"■ARE YOU READY TO WRITE ALL
OF THIS ON   THE TAPE?"
 2770 INPUT A$:IF A$="YES" THEN GOSUB 28
10
 2775 RETURN
 2800 REM WRITE TO TAPE
 2810 L=1:D=1:C=1:F$="DCALENDAR"
 2820 OPEN L,D,C,F$
 2825 IF E=1 THEN PRINT#L,NN:GOTO 2840
 2830 PRINT#L,N
 2840 FOR K=1 TO N
 2845 IF E=1 AND ST$(K)="FINISHED" THEN
2880
 2847 GOSUB 3000
 2850 PRINT#L,DA$(K)
 2860 PRINT#L,IT$(K)
 2870 PRINT#L,ST$(K)
 2880 NEXT K
 2890 CLOSE L:WR=WR+1:CH=WR
 2898 RETURN
 2899 REM SEARCH FOR DATE AND PRINT
 2900 CO=0
 2905 LET W=LEN(SE$)
 2910 FOR K=1 TO N
 2920 IF LEFT$(DA$(K),W)=SE$ THEN 2960
 2930 NEXT K
 2940 IF CO=0 THEN PRINT"■NOTHING FOR ";
SE$
 2950 RETURN
 2960 IF CO>0 THEN 2970
 2965 PRINT"■■■■THE ITEMS ON THE CALENDA
R ARE:"
 2970 PRINTTAB(3)DA$(K)
 2975 PRINTTAB(3)IT$(K)
 2980 PRINTTAB(3)ST$(K)
 2985 LET CO=CO+1
 2990 FOR Z-1 TO 1500:NEXT Z
 2995 PRINT"■":GOTO 2930
 3000 POKE59411,53:POKE 514,0:WAIT 514,1
6:POKE 59411,61:RETURN
```

IT$(N + 1). IT$ is the array that holds the items, and (N + 1) puts the new item in the correct location in the array. Finally, you enter the status of the item, whether it is finished or not, in line 2745 as ST$(N + 1).

In line 2750 the program updates N so that the correct number of items will be written on the tape. It also updates a variable CH, which counts the changes you make. Then you are asked if you want to add another item, and if you do you cycle through the whole thing again.

When you have finished adding items, you are asked if you want to write the items on tape. It is possible to write the data to tape either at this point or later when you are finished with the program.

There is one very satisfying thing about making lists: crossing all of those items off the list. STATUS is not quite as good as a heavy swipe across the page, but this routine, in lines 2605–2695, does let you do something rather like that. You can type "finished," a satisfying experience. It also is very useful in that it permits the operation of one of the search routines.

STATUS first asks for the date of the item to be changed. Since the user is not likely to remember the item number, you are given the option of looking at the items for the date. Then the user is asked to specify whether the new status is to become "finished" or "not finished." The loop in lines 2665–2675 searches the array in which the dates are stored to find the date specified, and when that date is found, it changes the value to the status array. Notice that the CH variable is also updated in this routine as it was in ADDITIONS.

Two search routines, TODAY and OTHER, operate in a similar way. TODAY converts the date entered at the beginning of the program to SE$, and then it goes to the subroutine beginning at line 2900. OTHER asks which date you would like to find, converts the answer to SE$, and then goes to the same subroutine. The search is done by the loop in lines 2910–2930. Statements 2905 and 2920 let you search for any part of the date. If you know that something is supposed to happen in April, but you do not remember the day, you can search for all entries in April by simply typing April, or its abbreviation, in answer to the question asking for the date to be found. If you know the exact date, April 28, for example, then you can search for that specific date. This feature also lets you abbreviate the month in any way you wish as long as you are consistent.

There is one slightly inconvenient side effect of this procedure for searching. If you ask for April 2, the search routine will not distinguish April 2 from April 22 or April 28. Thus, you need to use 01, 02 and 03 for indicating the first, second and third day of each month.

PAST is the routine that searches for unfinished items. Here is what is hanging over your head, as it says. Since this list may get long (at least mine does), there is a pause built in by line 2580 so that the whole list does not go scrolling off the screen faster than you can read it.

The routine that writes the data to the tape is in lines 2800–2899, and when combined with lines 980 through 990, these lines provide two special services for the user. After adding items to the calendar, the user can write these items to the data tape. In addition, when you are finished with the program, which is signified by typing "done," the program makes two checks.

The three arrays grow as the number of items added to the calendar increases. This means that the amount of memory used also increases. Line 980 checks to see if there are at least 200 bytes of memory left.

When there is less than this amount of memory, the program asks the user if he or she wants the program to write the tape deleting all of the finished items. If the answer is yes, line 985 sets the variable E to 1 and goes to the PAST routine. It lists all of the items that are un-

finished and counts them. Then it goes to the write routine, and the tape is written without the finished items, which should substantially reduce the memory used.

There is a funny-looking subroutine in line 3000 that needs to be explained. There is a glitch in the writing routines in the PET that does not usually pose a problem. But when writing some items and not writing others, this glitch produces funny (undesirable) results. Line 3000 is one "fix." It turns the tape drive motor off, waits for 16 jiffies and then turns the motor back on. It is needed to make sure your tape is copied accurately.

Lines 987–990 check to make sure that no changes not written on the tape have been made. In line 987 the program compares the values of CH and WR. If CH is less than or equal to WR, then any change that has been made has been put on the tape. If CH is greater than WR, the program asks the user if he or she wishes to add these changes to the tape. Even if you or I forget to rewrite the tape, the program will remind us that it needs to be done.

This program carries the user very naturally through its operation. There is, however, one feature that requires instruction. The first time the program is run, there is no data tape. If you start at the beginning of the program, it will ask you for a data tape that does not yet exist.

The solution to this apparent quandary is to run the program the first time with the command RUN 2190. This will ask "What's next?" and give a list of options. Use the ADDITIONS routine to enter several items for the calendar. Then when the program asks if you want to write these to tape, insert a new tape into the cassette, and you have a data tape. Henceforth you will be able to start at the beginning of the program.

Personable Qualities

There are four features of this program that make it a "personable" calendar. The first is variety. There is no reason why every statement by the computer or the user needs to be the same every time the statement is made. Isaac can say "hello" in four different ways. There is no reason that this strategy could not be used to take boredom out of the repetitious tasks.

Also, Isaac can look for and find my name no matter what sentence I embed it in. One version of this program had a subroutine that let the user say "yes" in any of five ways. There is little reason for the interaction of computer and human being to be stilted; at least not when a moderate amount of imagination is exercised.

Second, the program is self-describing. Type RUN 900 or ask for program, and the program describes itself. REMark statements are useful if the program is written on a piece of paper. But if the program is too large to fit on the screen for a single viewing, they are likely to be hard for the user to find.

Third, programs that learn can be written. Learning in the sense of accumulating, storing and using information is quite easy to build into programs. This program learns how many items there are on the data file, how many changes have been made and whether they have all been written on the tape, and whether it is approaching the bounds of memory. Short game programs that are likely to be played repetitively or teaching programs can easily be written so that they learn about the skill of the user as he or she plays the game.

Finally, the computer ought to be programmed to do as much of the work as possible. This program keeps track of two important aspects of its operation: changes and core availability. Then the user does not have to do this work with the consequent problems it entails.

For "personal" computers it would be nice to have "personable" computer programs.

There is one nice thing about finishing this article: now I can type "finished" by an item that has been hanging over my head for more than a month. ■

Recover That Lost Disk BASIC Program

A trick for the TRS-80.

Louise H. Frankenberg
1289 Magothy Rd.
Pasadena, MD 21122

If you have ever accidentally been rebooted back to DOS 2.1 while in the middle of keyboarding a long BASIC program, you will be happy to know that the program is still intact in memory and can easily be completely recovered. The following discussion assumes that you originally answered the FILES? question with the default value by merely pressing ENTER.

Storing the Program in Memory

Suppose you have typed in the following BASIC program:

```
1'AAAAA
2'BBBBBB
3'CCCC
```

Under DOS 'DEBUG' this will appear in memory as in Example 1.

The BASIC program always starts at location 68BA and ends after three 00 bytes in a row. The first two bytes of each BASIC statement contain the LSB and MSB of the starting address of the next statement. In the example, new statements start at 68BA, 68C7 and 68D5. The contents of address 68D5 point to an address containing 00, since that is the end of the program.

The next two bytes of each statement contain the BASIC line number (LSB and MSB). Here they contain 0100, 0200 and 0300. Next we have the BASIC code (3A 93 FB is TRS-80-ese for "remark," 41 = "A," 42 = "B" and 43 = "C"), and then a single 00 byte. The program end address that you'll need to know for program recovery is the address of the byte immediately following the three 00 bytes. In this example it's at 68E3.

```
68B0   FF 00 FF 00 FF 00 FF 00   FF 00 C7 68 01 00 3A 93
68C0   FB 41 41 41 41 41 00 D5   68 02 00 3A 93 FB 42 42
68D0   42 42 42 42 00 E1 68 03   00 3A 93 FB 43 43 43 43
68E0   00 00 00 FF    (etc.)
```

Example 1.

Program Recovery

Now that we've seen how the program is stored in memory, here are "cookbook" instructions for recovering your program (values for the example are in parentheses).

1. Go into the 'DEBUG' program immediately upon finding yourself in DOS.
2. Display the 6800 page and jot down the contents of addresses 68BA and 68BB (C7 and 68); convert to decimal (199 and 104).
3. Page through memory with 'DEBUG' until you find three 00 bytes in a row and jot down the *following* address (68E3); convert to decimal (LSB E3 = 227 and MSB 68 = 104).
4. Enter G5200 to return to BASIC (*never* use G402D and reload BASIC, or you'll wipe out the start of the program).
5. Answer FILES? and MEMORY SIZE? as originally.
6. Tell the computer what's in the beginning of the program and where to find the end. For the example you would POKE the following:

```
POKE 26810,199  (contents of 68BA)
POKE 26811,104  (contents of 68BB)
POKE 16633,227  (LSB of end address)
POKE 16634,104  (MSB of end address)
```

That's all there is to it! Your program is now ready to run, save on disk, add to or whatever.

Additional Notes and Comments.

I worked out the above the hard way. As a newcomer to disk, I lost hours of program typing in the process. An accidental reboot to DOS when doing disk I/O can be avoided in the first place by disabling the clock interrupt: either use CMD"T" or use 'DEBUG' to change addresses 46B8–46BF to the following (the latter is Radio Shack's new official patch):

CB 57 20 13 FE 20 28 11

If you originally answer the FILES? question with something other than the default value, your program will start at a different address (it's at 69DC for FILES?4). Find it with 'DEBUG'.

You can also recover a program that has "disappeared" after entering 'NEW'. In that case you would use CMD"S" to return to DOS, and go into 'DEBUG' as before. Since 'NEW' wipes out the contents of 68BA and 68BB, you will have to find the address where the second BASIC statement starts (68C7, in the example) and jot it down to poke into 68BA and 68BB. All the remaining steps are identical.

One more note: After I wrote this article, Radio Shack released DOS 2.2, which contained the clock-interrupt disable patch. This cured the problem for me. 2.2 appears to contain new bugs for multiple-drive use, but if you have only one disk drive I recommend you pick up your free copy as soon as possible. ■

METAMORPHOSIS!

Now you can add high resolution graphics to your Commodore PET computer. The MTU K-1008-6 GRAPHIC INTERFACE can be used with either old, new, or business PET computers. It is simple to use, and fits inside the PET for protection.

The GRAPHIC INTERFACE gives you easy control over each dot in a matrix which is 320 wide by 200 high for a total of 64,000 dots. Because each dot can be controlled, either graphic images, text lines, or any mixture of the two can be displayed. Since each dot is controlled from software you can even design your own special character font or graphic image set (logic, chemical, architectural).

INTERFACE TO ALL PETS - With separate connector boards for each style PET (K-1007-2 for OLD PETS, K-1007-3 for NEW). The K-1008-6 can be used with either.

THREE TYPES OF VIDEO - You can select either normal PET video, graphic video, or the COMBINED image of both video signals simultaneously!

8K RAM MEMORY EXPANSION - The graphic matrix requires 8K RAM which is supplied onboard. This memory can be used for program or data storage when not being used for graphics (or see your program in binary on the display!).

OLD OR NEW PETS CAN NOW HAVE HIGH RESOLUTION GRAPHICS

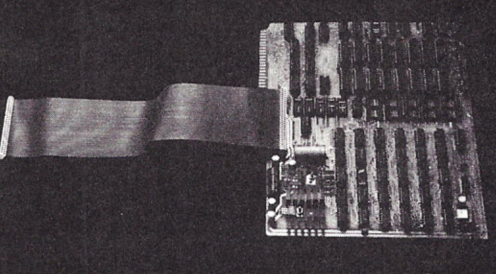
FLEXIBLY ADDRESSED ROM SOCKETS - Five ROM sockets are included on the board. They can be set *at the same* or different addresses, with you controlling which sockets are enabled at any time through software control. You also choose the sockets to be enabled when the PET is turned on.

EXTERNAL EXPANSION - This board also creates the KIM memory expansion bus supported by all MTU products. This allows insertion into our K-1005-P card file for expansion up to 4 other boards outside the PET case.

LIGHT PEN - The board has been designed to work with an optional light pen which MTU will be announcing soon.

SOFTWARE INTERFACED TO BASIC - MTU also has available machine language software to allow you to plot points, draw lines, and display characters at high speed.

Call or write for our full line catalog of products.



MICRO TECHNOLOGY UNLIMITED
P.O. Box 12106
2806 Hillsborough Street
Raleigh, N.C. 27605
(919) 833-1458

Reader Service No. 154

MTU
Micro Technology Unlimited
P.O. Box 12106
2806 Hillsborough Street
Raleigh, N.C. 27605

CP/M Is for Me

A favorable look at this controversial system.

Ken Barbier
PO Box 1253
Borrego Springs, CA 92004

In "CP/M and You" (*Microcomputing*, February 1980, p. 183) Thom Hogan voiced some complaints about CP/M while making some valid points both for and against the operating system. But the overall picture he painted was a little too gloomy. CP/M is, I think, the second most valuable piece of software for either mini- or full-sized floppy disks, running second only to Microsoft Disk Extended BASIC.

I have been running a single-drive CP/M system on 5¼-inch single density disks for a year now and haven't experienced the insufficient disk space that Hogan describes. The secret is to eliminate from your working disks all of the CP/M-supplied transient programs that you won't use in day-to-day operation. See Listing 1 for a record of a session that prepares a "packed" system disk for use in writing assembly-language programs.

Start with a Fresh Copy

A new CP/M user must first write-protect the system disk and make copies of it. You do this on a single-drive system by using routines supplied with your disk operating system (DOS). Next, you should use MOVCPM and SYSGEN to generate the largest version of the operating system you can fit into your available read-write memory. You then won't need these programs on the disks you will be using daily. Other programs supplied on your system disk can also be eliminated on your working disks. But before you start erasing, make your system back-up copies.

You are now ready to take one of these copies and use it to generate a "packed" system disk. This will take the better part of an hour. Then you can write-protect this disk and make copies of it for your day-to-day use.

Packing a system disk involves more than erasing the unwanted files. This would leave a fragmented disk directory full of small holes where the old programs have been erased. Your new, long files would be broken into little pieces and scattered over the disk to fill all the holes left by erasing the unneeded files. This would cause an unnecessary amount of track-to-track stepping and too much waiting for the next sector to come around.

Why Do You Need to Pack?

Some operating systems for floppy disks will only write contiguous files. In such systems, each file has a starting track and sector and a file length recorded in the directory. On the disk, each sector of the file immediately follows the previous sector, until the complete file is written.

When files are deleted from the directory, that space is not used by the next write operation, because it might not be large enough to permit writing the entire file in one contiguous block. In such a system, periodic packing sessions are required following file erasures or updates. Packing a disk will move all the remaining files down in the disk address space to fill in all the holes.

This system's advantage is that once your disk drive read-write head has found

Even with only a single mini-floppy disk drive, the CP/M operating system can provide a comfortable solution to mass storage. But some things are inevitable, so room has been left in the cabinet for a second drive.

the first record of a file, it need only read consecutive sectors and tracks until the entire file is in memory. This results in the fastest possible load time. The disadvantage of the system is that in a typical work session involving repeated edit and assembly operations, the disk space will fill up more quickly, and you will have to stop while the system packs the disk.

CP/M Does It Smarter

CP/M uses a different technique, employing a disk map. The entire disk address space (track n, sector m) is mapped in the directory; a block of 1024 (1K) is the minimum size. In the soft and 16-sector formats, each sector contains 128 bytes, so each map entry points to a block of eight sectors. In the North Star ten-sector format, each entry addresses four 256-byte sectors (the double-density scene is even more complicated).

Each map entry points to the track and sector number of the first sector of each block only, so the remaining sectors must immediately follow each other. But this is true only within each 1K block, not the entire file, as was true with the other system.

This mapping means that when we write a long file, CP/M will find the first available 1K block on the disk and fill it, then find the next block and fill it and so on. If we have erased a number of short files—for example, one on track 6, one on track 14 and one on track 23—our long file will be broken up into 1K blocks and stored in all the holes on the disk. Loading a file that has been scatter-written takes considerably longer, and if your drive uses a noisy stepper motor, you can hear the numerous track-to-track seeks. Such drives sound nervous, buzzing from track to track collecting the program.

A disk-mapped operating system's advantage is that the disk takes longer to fill up, since every available block is filled from the bottom of the disk up. A disadvantage, in addition to possible fragmentation, is that no file can be less than 1K. The last block assigned to a file can possibly contain only a single byte, but it still takes up 1K of disk space!

We should not simply erase the unwanted files on our CP/M disk and start using the disk at that point. We should first pack the disk. The session that produced the accompanying listing is not something you would want to endure very often, so go through it only once, and then make copies of the resulting packed system disk for your daily operations.

Start Packing

The listing starts with a dump of the directory of the CP/M for North Star disk as supplied by Lifeboat Associates. In this example, MOVCPM and SYSGEN have been renamed, since they contain procedures specific to the North Star computer. Assuming we have completed our system generation and CBIOS (Custom BASIC Input-Output System), we are ready to pack.

First, decide what programs you want to save and put them in order according to frequency of use. Since you are going to speed up your disk access by packing a system disk, you might as well have the most-used programs at the start of the disk, where they can be found fastest.

Working in assembly language, you are going to be editing, assembling, debugging, editing, assembling and debugging. Thus, you want to pack your disk in the following sequence: ED.COM, ASM.COM, DDT.COM. What comes next is pretty much a toss-up. Note that I have erased SUBMIT.COM. You might want to hang on to it, unless you are sure you won't be using it.

You have listed the directory and used STAT to see how much space each file occupies and how much disk space you have left. To pack the disk, you will have to move all the programs you want to save to the top of the disk address space and then clear out the bottom of the disk space to produce one large hole. When you then move the saved programs back down, they will each

```
A>DIR
A: NRELOC    COM
A: STAT      COM
A: PIP       COM
A: SUBMIT    COM
A: ED        COM
A: ASM       COM
A: DDT       COM
A: LOAD      COM
A: NSYSGEN   COM
A: DUMP      COM
A: DUMP      ASM
A: USER      ASM
A>STAT *.*

RECS BYTS EX D:FILENAME.TYP
  64   8K  1 A:ASM.COM
  38   5K  1 A:DDT.COM
  32   4K  1 A:DUMP.ASM
   4   1K  1 A:DUMP.COM
  48   6K  1 A:ED.COM
  14   2K  1 A:LOAD.COM
  78  10K  1 A:NRELOC.COM
   8   1K  1 A:NSYSGEN.COM
  55   7K  1 A:PIP.COM
  24   3K  1 A:STAT.COM
  10   2K  1 A:SUBMIT.COM
  14   2K  1 A:USER.ASM
BYTES REMAINING ON A: 27K


A>NOW LET'S SAVE SOME SPACE
NOW?

A>ERA *.ASM
A>PIP ED.SAV=ED.COM

A>PIP ASM.SAV=ASM.COM

A>PIP DDT.SAV=DDT.COM

A>PIP STAT.SAV=STAT.COM

A>PIP PIP.SAV=PIP.COM

A>PIP LOAD.SAV=LOAD.COM

A>PIP DUMP.SAV=DUMP.COM

A>STAT
A: R/W, SPACE: 1K


NOW WE'LL MOVE EVERYONE BACK
NOW?

A>ERA *.COM
A>REN PIP.COM=PIP.SAV
A>PIP ED.COM=ED.SAV
A>PIP ASM.COM=ASM.SAV

A>PIP DDT.COM=DDT.SAV

A>PIP STAT.COM=STAT.SAV

A>PIP PIPX.COM=PIP.COM

A>PIP LOAD.COM=LOAD.SAV

A>PIP DUMP.COM=DUMP.SAV

A>ERA *.SAV
A>ERA PIP.COM
A>REN PIP.COM=PIP..X.COM
A>NOW LET'S SEE WHAT WE'VE GOT
NOW?

A>DIR
A: ED        COM
A: ASM       COM
A: DDT       COM
A: STAT      COM
A: PIP       COM
A: LOAD      COM
A: DUMP      COM
A>STAT *.*

RECS BYTS EX D:FILENAME.TYP
  64   8K  1 A:ASM.COM
  38   5K  1 A:DDT.COM
   4   1K  1 A:DUMP.COM
  48   6K  1 A:ED.COM
  14   2K  1 A:LOAD.COM
  55   7K  1 A:PIP.COM
  24   3K  1 A:STAT.COM
BYTES REMAINING ON A: 46K

A>NOW PATCH <313E+BIAS> FOR
40 TRACKS NOW?

<AT THIS POINT THE CONTROL PANEL
 IS USED TO START A MONITOR
 PROGRAM IN PROM>

-S713E
713E 4F 5B
713F C0 .
-GA040

<THE ROUTINE IN PROM AT ADDRESS
 A040 WRITES TRACKS 0 - 2 FROM
 CP/M IN MEMORY>

A>NOW HOW MUCH DO WE HAVE?
NOW?

A>STAT
A: R/W, SPACE: 58K
```

Program listing. Console messages during a mini-disk packing session. Eliminating unneeded files from the CP/M system disk increases the available workspace by 19K. Packing the remaining programs reduces access time. A final system patch permits the use of 40-track disk drives, adding another 12K.

be written in one contiguous block.

Using the STAT listing, you have 27K of space on the disk, and the programs you want to save total 32K. So you start by erasing the .ASM files. Next, PIP is used to move the seven files you want to save. Since you can't use duplicate names, tell PIP to make a file copy of each program in turn, but with a file type of .SAV.

Once you have your seven .SAV files, STAT tells you that you have used up all of the disk except a single 1K block. One side benefit of this procedure is that it gives the operator, PIP, and the disk drive a real test and checks out the whole disk surface in the process.

Now Do It All Over Again

Now you create your big hole at the bottom of the disk space by erasing all of the .COM files (are you sure you have everything backed up?). You need a PIP.COM file to use for the move back, so you next have to rename PIP.SAV as PIP.COM. This new PIP is then used to move the .SAV files back down as .COM files. In the process, PIP has to move itself, so you temporarily call this new PIP file by another name.

When you are done with the packing, you erase PIP.COM and rename the packed version of PIP. Now DIR and STAT *.* are used to inspect the new packed system disk. You have 46K of workspace, a reasonable amount for an assembly-language programmer. Since you are talking about a single-drive mini-floppy CP/M system, you should keep in mind that you don't want to get too much work on a single disk anyway, since the only way you have to copy a file is to copy the whole disk.

I have found that this size disk is convenient for assembly-language work, my primary use for the system. A couple of reasonably sized source files and their hex, object and print files will pretty well fill a disk. At that time, it is a good idea to file that disk away for safekeeping.

BASIC is another matter, since BASIC language programs breed like rabbits, but on a single drive system you need only save BASIC itself, and maybe STAT. If you erase everything else on the disk, you will end up with about 40K of workspace.

Even More Space

I ran across a real bargain in Wangco Model 82 mini-floppy drives, and since they can access 40 tracks instead of the standard 35, I had to figure out a method for patching CP/M to make it use the extra track. At the bottom of the listing you will see that I found and patched the location in CP/M that tells it how much disk space is available.

Five more tracks add $5 \times 10 \times 256$ (12,800) bytes in the North Star format, so I incremented the 4F (hex) by C (hex), which adds an extra 12K to the usable disk space. You will have to be certain that your DOS will properly address 40 tracks before making this patch, however. And note that the GA040 accesses a "write tracks 0 through 2 only" routine in my DOS. Yours will not be at this address.

Comfortable Conclusions

The Wangco drives also permit the use of both sides of each disk, which should be called flippy-disks to avoid confusion with read double-sided-type drives, which have two heads. This combination of two sides and 40 tracks makes a very usable configuration. I'm not anxious to tackle the problems of double density because I find the present configuration reliable and comfortable. Eight-inch floppies are a bit awkward to handle. The stiffer mini-floppies store well, and CP/M is a good operating system for them.

However, even this system needs a single-drive filecopy routine. I have written one, and next month's issue will have a complete assembly-language listing. ■

Some Notes on Termination

Protect your program from those incessant glitches.

Reo W. Pratt
2264 Cambridge Dr. SE
Grand Rapids, MI 49506

```
D400  C3  JMP
D401  00  D400
D402  D4
```

Listing 1. "Jump to here" program.

The ad said, "Eliminates noise, ringing, cross talk and overshoot!" It claimed that if you had any inexplicable problems, such as your system running fine and suddenly "crashing," the cause might be an unterminated motherboard. Proper termination will eliminate those frustrating "glitches" that appear only long enough to destroy the program you keyed in.

I recently had the opportunity to verify these claims with the help of Bill Godbout and two of his technical staff. Although most computerists accept the need for bus termination (most motherboards come with onboard termination now), you might be interested in the results of the tests I ran, particularly if you own or are planning to buy one of the older S-100 computers with an unterminated motherboard.

The Reason for Termination

When a signal or pulse is sent out on a bus line, it has to go somewhere. Many pulses placed on the bus by the processor are not used by other devices plugged into the board. When these pulses reach the end of an unterminated line, they bounce back, induce signals in adjacent lines, radiate into space—or some combination of all three.

The degree to which this occurs depends on several interrelated factors. The most significant are the signal frequency, the length of the conductor and the spacing between adjacent conductors. The higher the frequency, the more critical proper termination becomes. Even the slowest microprocessors deal with frequencies well into the radio-frequency spectrum. If you have trouble accepting this, remember that the clock fre-

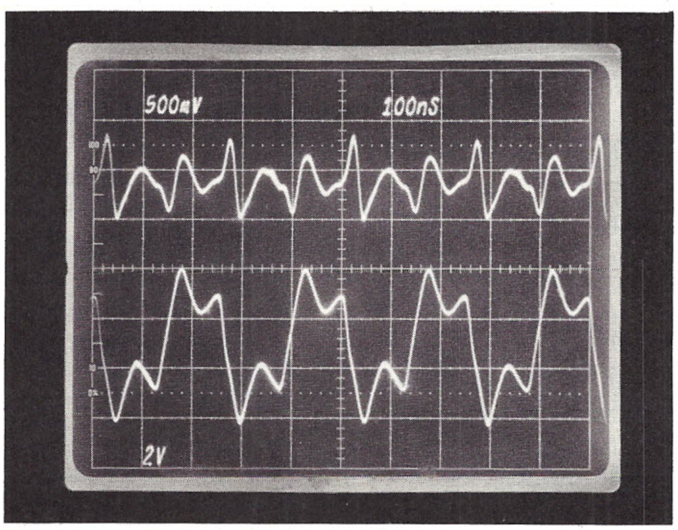


Photo 1. $\overline{POC}$ and $\overline{CLOCK}$ lines with active termination enabled.

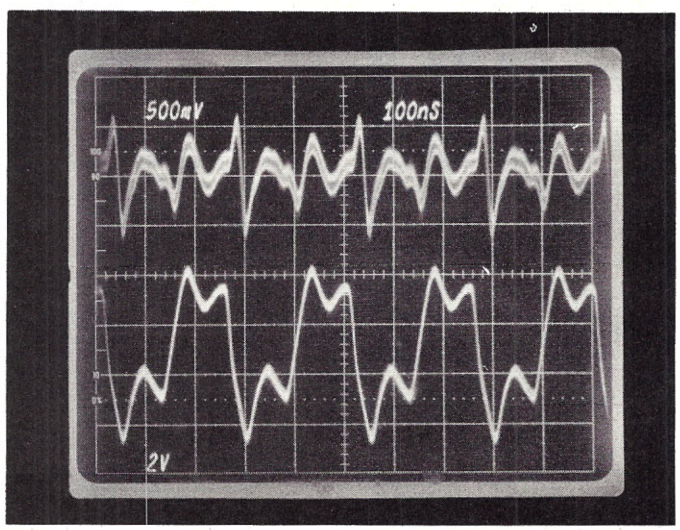


Photo 2. $\overline{POC}$ and $\overline{CLOCK}$ lines without termination.

quency of an 8080 is 2 MHz. This is much higher in the spectrum than the highest point on your AM dial (1.6 kHz).

Termination Problems

Noise is electrical energy on the bus that shouldn't be there. It comes from a variety of sources and, once it appears, won't go away—until you start to look for it.

Ringing occurs when a signal that is changing logic level (pulled high or low) doesn't stay at its intended level. This appears as a damped oscillation in the photographs.

Cross talk occurs when a signal on one line is imprinted on the line or lines adjacent to it. It causes confusion when the altered signal arrives at its destination.

Overshoot happens when a signal changes logic state with such force that it "shoots" past its intended level. It is closely associated with, and precedes, ringing.

To observe these problems, I used a Tektronix oscilloscope with 50 MHz bandwidth and dual-trace capability. I observed the various signal lines on an "older" Imsai 8080 with a Godbout active-terminator board that unplugged to look at the terminated and unterminated lines. I recorded the oscilloscope pictures with a 4×5 Graphex camera with a Polaroid back.

To keep the processor continually cycling during the test, I keyed in a short "jump to here" program (see Listing 1).

Photo 1 shows POC (top) and CLOCK lines. It is an excellent example of cross talk. Note how the POC line appears to follow the CLOCK line up and down. Note also that the POC voltage scale is set to ¼ the value of the CLOCK line—500 millivolts, as compared to two volts for the CLOCK trace. If the voltage scales were the same, the variations in the upper trace would not be as pronounced.

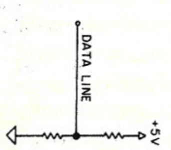


Fig. 1. Passive termination.

The terminator board improves the situation. Photo 1 shows the terminating network enabled. Photo 2 shows the effects of no termination. Noise, cross talk, undershoot and overshoot are all present in ample quantities. Compare the same portions of both traces. As you can see from the scope's illuminated graticule, the camera was well focused. The thickness of the traces, particularly the upper one, is due to noise, not poor photography.

Photos 3 and 4 show the PDBIN line in the upper trace and the DI7 line in the lower trace. The terminator board is enabled in Photo 3; it is removed from the machine in Photo 4. These photos show best how the terminator board reduces overshoot and ringing. Using two volts per division as shown on the scope face, you can see that the high logic level with the terminator in is only about three volts, whereas it is over four volts with no termination.

Technical Aspects of Termination

Amateur radio operators are familiar with the principles of termination through antenna and transmission-line theory. The bus lines in a computer carry a signal from one point to another as does a transmission line. The terminating network in a computer does more than just provide a proper load to sink signal current. It also sources current for logic signals that may lack sufficient drive and holds the logic high voltage at the optimum level.

Ones and zeros are generally represented by five and zero volts, respectively. However, a look at a data sheet shows that most TTL gates view anything over two volts as logic high, with 2.6 volts optimum. Holding the logic-high-state voltage to the optimum reduces overshoot since the signal only has half as far to go when it changes state.

Fig. 1 shows the simplest form of termination: two resistors for each line wired to the 5 volt "rail" and ground to source and sink current as required. It is effective but has one drawback. With all those resistors wired between +5 volts and ground for nearly 100 lines, the power supply has an increased load to bear. If you apply Ohm's law, you'll find that the extra load is about ½ Amp. This may not be too much, depending on the system, but there is a better way.

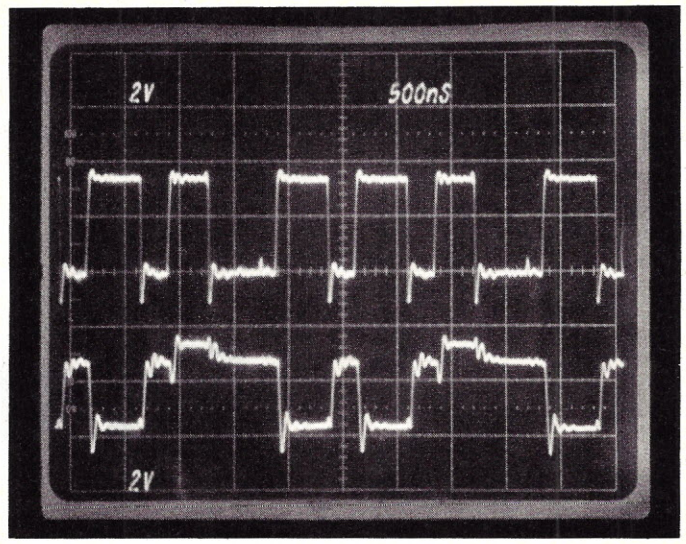


Photo 3. PDBIN and DI7 lines. Termination enabled.

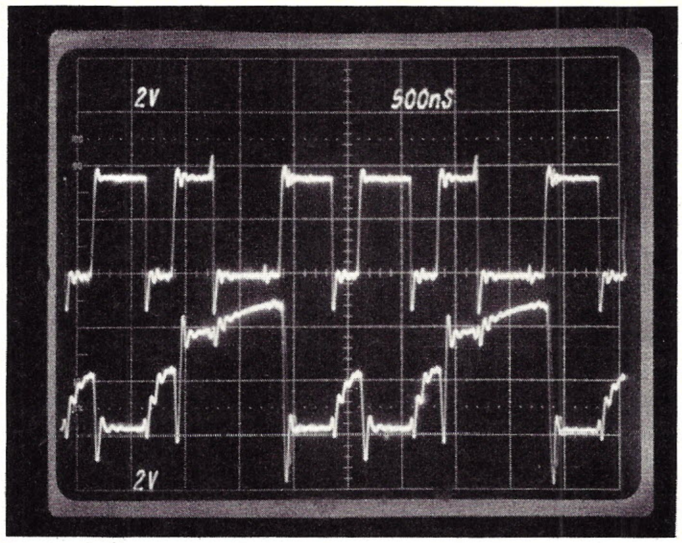


Photo 4. PDBIN and DI7 lines. Terminator removed from bus.

Active termination has all the advantages of passive termination but draws far less current from the power supply and costs the same or less. Active termination uses only one resistor per line instead of two and a sensing network composed of an operational amplifier that sinks or sources current only as it is needed. It effectively "switches" the terminating network on and off the bus lines as required, eliminating the constant power draw. Total constant current draw on the Godbout active terminator board, for example, is only 20 milliamperes.

Proper termination of the lines on a motherboard or backplane is simply a design necessity that never should have been omitted from the original S-100 systems. Recently advertised motherboards from Godbout have Faraday shielding between the lines as well as active termination networks on the board. Other manufacturers also advertise similar boards. Look into the "termination situation" carefully. It could save a lot of frustration in your computing future. ■

Fastfind

For the computerist in a hurry to search large arrays.

Bill Roch
24000 Bessemer St.
Woodland Hills, CA 91367

This article describes a fast way to find a match between a variable and a variable in an array. One way to do this is to use a number of IF statements (see Example 1).

This is fine for checking a couple of variables, but why write the same thing over and over again when you can do it with a FOR-NEXT loop in three or four statements? The FOR-NEXT loop is a better way to find a match between a single variable and a variable in a small array (see Example 2).

This is an excellent method if the array is small or the program only executes this loop a few times. However, if the array contains 1000 variables or the program uses this loop for every record it processes, the computer will spend a lot of time "spinning its wheels," i.e., comparing nonmatching variables.

The same waste of time occurs when the computer searches for the matching index or key in a sequential random file. Most of the computer time is spent asking, "Do I have a match?" and answering "No—Go get the next record."

With a FOR-NEXT loop, the average search will be made through half of the array to find a match—all wasted time except for the final match.

```
10 IF A=B(1) GOTO 100
20 IF A=B(2) GOTO 200
 .
 .
100 IF A=B(10) GOTO 1000
```

Example 1.

```
10 FOR K=1 TO 10
20 IF A=B(K) GOTO 50
30 NEXT K
40 PRINT "ERROR": STOP
50 ON K GOTO 100, 200, . . . . 1000
```

Example 2.

A Faster Way

Using the FASTFIND routine on an array of 1000 sequential variables will only require a maximum of ten tries to find any variable in the array.

To find 999 in an array of 1000 the first attempted match would be on 500. Since the array variable of 500 is less than the 999 being looked for, the next attempt is made at 750. The next attempt would be at 875. Each time the area to searoh is cut in half. This continues until the match is found.

Table 1 shows the calculated subscript used to find a match for the number 955 in an array of 1 to 1000. The low subscript number is set to 0, and the high is set to 1001. As before, the first attempted match is made at 500—low + high divided by two. The value 955 is higher than 500 so the 500 is used as the low subscript number and the process is repeated.

This continues until the 955 is lower than the calculated subscript number, at which time the calculated subscript is substituted for the high subscript number. Decimal numbers are rounded down (turned into integers) so that one half of (938 + 1001) equals 969, not 970. On the ninth iteration the number 955 matches the subscript.

Low subscript number	High subscript number	Low & High divided by 2	Number related to subscript
0	1001	500	Higher
500	1001	750	Higher
750	1001	875	Higher
875	1001	938	Higher
938	1001	969	Lower
938	969	953	Higher
953	969	961	Lower
953	961	957	Lower
953	957	955	Match

Table 1. Locating the subscript for 955.

Why is the low-high range 0 to 1001, while the actual range to be searched is from 1 to 1000? This is done so the subscripts 1 and 1000 will find a match. Otherwise, we get 1+2=3 and half of 3=1, so subscript 2 is never found. At the other end we have 1000+999=1999, and half of 1999=999—another endless loop.

Routines

Two sample routines are provided; each has a test driver program for testing the routine. Listing 1 is in standard BASIC, and Listing 2 is in Tarbell TBASIC, which makes use of some of the TBASIC features. Both programs do the same thing. Listing 1 fastfinds a number, while Listing 2 fastfinds a string.

Listing 1 illustrates FASTFIND for a number match. The program first builds an array containing 1000 variables ranging from 1 to 3001 using every third number (1,4,7, . . . 3001). The printed output from this program shows the count of the tries it took to find a match for the entered variable. When the number entered from the keyboard is out of range, the program requests that a correct number be reentered.

When the number entered from the keyboard is not found,

```
F A S T F I N D   R O U T I N E

ENTER ARRAY VALUE:? 1
 9 TRYS - ARRAY VALUE [ 1 ] IN A(1)

ENTER ARRAY VALUE:? 3
 10 TRYS - ARRAY VALUE [ 3 ] NOT FOUND
   NEXT LOWEST ARRAY VALUE IS [ 1 ] IS IN A(1)

ENTER ARRAY VALUE:? 4
 10 TRYS - ARRAY VALUE [ 4 ] IN A(2)

ENTER ARRAY VALUE:? 3001
 8 TRYS - ARRAY VALUE [ 3001 ] IN A(1001)

ENTER ARRAY VALUE:? 3000
 10 TRYS - ARRAY VALUE [ 3000 ] NOT FOUND
   NEXT LOWEST ARRAY VALUE IS [ 2998 ] IS IN A(1000)

ENTER ARRAY VALUE:? 2998
 10 TRYS - ARRAY VALUE [ 2998 ] IN A(1000)

ENTER ARRAY VALUE:? 1504
 1 TRYS - ARRAY VALUE [ 1504 ] IN A(502)

ENTER ARRAY VALUE:? 1000
 9 TRYS - ARRAY VALUE [ 1000 ] IN A(334)
```

Sample standard BASIC run.

```
Enter Array Value: AA00
 9 Trys - Array Value [AA00] is in ARRAY$(1)

Enter Array Value: EJ90
 1 Trys - Array Value [EJ90] is in ARRAY$(500)

Enter Array Value: JJ90
 10 Trys - Array Value [JJ90] is in ARRAY$(1000)

Enter Array Value: AA10
 10 Trys - Array Value [AA10] is in ARRAY$(2)

Enter Array Value: AA01
 10 Trys - Array Value [AA01] NOT FOUND
   Next Lowest Array Value is [AA00] in ARRAY$(1)

Enter Array Value: AA09
 10 Trys - Array Value [AA09] NOT FOUND
   Next Lowest Array Value is [AA00] in ARRAY$(1)

Enter Array Value: BA40
 8 Trys - Array Value [BA40] is in ARRAY$(105)
```

Sample Tarbell BASIC run.

```
10 REM ****************************************
20 REM **                                    **
30 PRINT "F A S T F I N D   R O U T I N E"
40 REM **            TEST PROGRAM            **
50 REM **                 BY                 **
60 REM **              BILL  ROCH            **
70 REM ** Standard Basic                     **
80 REM ****************************************
90 REM
100 DIM A(1004)
110 J=0
120 FOR K=1 TO 3002 STEP 3:REM  BUILD AN ARRAY
130 J=J+1
140 A(J)=K:REM                  RANGE 1 TO 3001
150 NEXT K:REM                  1,4,7.....3001
160 L1=0:REM                    LOWEST SUBSCRIPT VALUE MINUS 1
170 H1=J+3:REM                  HIGHEST SUBSCRIPT VALUE PLUS STEP
180 PRINT :INPUT "ENTER ARRAY VALUE:? ";B
190 IF B<=L1 OR B>K-3 THEN GOTO 160:REM        OUT OF RANGE
200 GOSUB 400
210 PRINT C1;"TRYS - ARRAY VALUE [";B;"] ";
220 ON E1 GOTO 250
230 PRINT "IN A(";STR$(K1);")"
240 GOTO 160
250 PRINT "NOT FOUND"
260 PRINT "   NEXT LOWEST ARRAY VALUE IS [";A(K1);"]";
270 PRINT " IS IN A(";STR$(K1);")"
280 GOTO 160
290 REM
300 REM ****************************************
310 REM **                                    **
320 REM **  F A S T F I N D   R O U T I N E   **
330 REM **                                    **
340 REM ****************************************
350 REM
360 REM INPUT TO ROUTINE
370 REM               B  - VALUE TO FIND IN ARRAY A(X)
380 REM               L1 - LOWEST ARRAY SUBSCRIPT
390 REM               H1 - HIGHEST ARRAY SUBSCRIPT
400 REM OUTPUT FROM ROUTINE
410 REM               C1 - NUMBER OF TRYS COUNT
420 REM               E1 - ERROR FLAG
430 REM                    0 = FOUND
440 REM                    1 = NOT FOUND
450 REM               K1 - SUBSCRIPT NUMBER IN ARRAY
460 REM                    CONTAINING FOUND VALUE
500 E1=0:REM                          RESET NOT FOUND FLAG
510 C1=0:REM                          ZERO FIND COUNTER
520 K1=INT((L1+H1)/2):REM             CALCULATE FIND NO.
530 IF K1<>L1 THEN GOTO 560:REM       IF CALC NO = LOW NO - NOT FOUND
540 E1=1:REM                          SET NOT FOUND FLAG
550 GOTO 640
560 C1=C1+1:REM                       ADD 1 TO TRY COUNTER
570 IF B=A(K1) THEN GOTO 640
580 IF B>A(K1) THEN GOTO 600
590 IF B<A(K1) THEN GOTO 620
600 L1=K1:REM                         MAKE LOW SUBSCRIPT = POINTER
610 GOTO 520
620 H1=K1:REM                         MAKE HIGH SUBSCRIPT = POINTER
630 GOTO 520
640 RETURN
```

Listing 1. FASTFIND routine in standard BASIC.

the program sets a NOT FOUND flag and prints out NOT FOUND. It also prints out the next lowest array value and its subscript position in the array.

The Tarbell BASIC program (Listing 2) illustrates how the FASTFIND routine could be used in finding selected sequential keyed records in a random file. The test program first creates an array containing string values, simulating a sequential set of record keys.

These keys (or index) range from AA00 through JJ90 (AA00,AA10,AA20 . . . AB00. . . JJ90). This string array could consist of names, part numbers, account numbers, policy codes, etc. The program using the FASTFIND routine would first read all the keys or IDs into an array similar to the one generated by the program.

The routine in this program operates in the same manner as the standard BASIC routine. It keeps cutting the search range in half until a match is found. When the match is found, the subscript number becomes the record number for the file read from a random file.

If no match is found, the routine will return an error flag that causes the NOT FOUND to be printed. The next lowest array value with the subscript of that value is also printed.

It may be that 80 percent of the time the desired record is in the top (high subscript values) end of the file. Instead of going in at the middle of the file and

calculating all the way to the number, go in just below where most of the action is. This will save two or three extra attempted matches.

Suppose the routine is being used with a name file that keys on names. In the case of a filename with five JONESes, the routine will return as soon as it finds a JONES. There is one chance in five that it found the right JONES, and it will always find the same JONES and return the same subscript value. By adding to or subtracting from the subscript, the desired record can be found.

Another way would be to subtract five from the subscript and use a FOR-NEXT loop to find a match on JONES and JONES' first name.

Three nice features of Tarbell's TBASIC are shown in Listing 2:

- Meaningful line labels instead of line numbers.
- Long variable names.
- Ability to transfer variables with the GOSUB, RETURN and RECEIVE statements.

Summary

Considerable computer time can be saved using a FASTFIND routine instead of a FOR-NEXT loop when searching large arrays. A routine of this type can work equally well with strings and numbers. The number of reads required to find the proper key in a random file can be greatly reduced.■

```
TSTPROG REM ****************************************
        REM **                                    **
        REM **  F A S T F I N D   R O U T I N E  **
        REM **          TEST  PROGRAM             **
        REM **               by                   **
        REM **           BILL  ROCH               **
        REM ** Tarbell Basic                      **
        REM ****************************************
        REM
START   DIM ARRAY$(1001):       FALSE=0:        TRUE=1
        CNTR=0:                 ZERO=0
        FOR K1=1 TO 10: A$=CHR$(64+K1):REM      BUILD TEST ARRAY
        FOR K2=1 TO 10: B$=CHR$(64+K2):REM      AA00,AA10,AA20....
        FOR K3=1 TO 10: CNTR=CNTR+1:REM         AA90,AB00....JJ90
        ARRAY$(CNTR)=A$+B$+STR$(K3-1)+"0"
        NEXT K3,K2,K1
TSTLOOP PRINT :INPUT "Enter Array Value: ";INPUT$
        IF LEN(INPUT$)<>4 THEN PRINT "NOT 4 CHARACTERS":GOTO TSTLOOP
        IF INPUT$<"AA00" OR INPUT$>"JJ90" THEN GOTO TSTLOOP
        LOW=ZERO:               HIGH=CNTR+1
        GOSUB FASTFIND,LOW,HIGH,INPUT$
        RECEIVE ERRFLG,TRYCOUNT,SSCRIPT
        PRINT TRYCOUNT;"Trys - Array Value [";VALUE$;"] ";
        ON ERRFLG GOTO FOUND
        PRINT "NOT FOUND"
        PRINT "   Next Lowest Array Value is [";ARRAY$(SSCRIPT);"]";
        PRINT " in ARRAY$(";STR$(SSCRIPT);")"
        GOTO TSTLOOP
FOUND   PRINT "is in ARRAY$(";STR$(SSCRIPT);")"
        GOTO TSTLOOP
        REM
        REM ****************************************
        REM **                                    **
        REM **  F A S T F I N D   R O U T I N E  **
        REM **                                    **
        REM ****************************************
        REM
        REM     INPUT TO ROUTINE
        REM             BOTTOM - LOWEST ARRAY SUBSCRIPT
        REM             TOP    - HIGHEST ARRAY SUBSCRIPT
        REM             VALUE$ - VALUE TO FIND IN ARRAY ARRAY$(X)
        REM     OUTPUT FROM ROUTINE
        REM             FLAG   - ERROR FLAG
        REM                  FALSE = NOT FOUND
        REM                  TRUE  = FOUND
        REM             COUNT  - COUNT OF TRYS
        REM             POINTER- SUBSCRIPT NUMBER IN ARRAY
        REM                      CONTAINING FOUND OR NEXT
        REM                      LOWEST VALUE
        REM
FASTFIND        RECEIVE BOTTOM,TOP,VALUE$
        FLAG=TRUE:      COUNT=ZERO
SETPNTR POINTER=INT((BOTTOM+TOP)/2):REM         CALC FIND NUMBER
        IF POINTER<>BOTTOM THEN GOTO FIND
        FLAG=FALSE:     GOTO RETURNX
FIND    COUNT=COUNT+1
        IF VALUE$=ARRAY$(POINTER) THEN GOTO RETURNX
        IF VALUE$>ARRAY$(POINTER) THEN BOTTOM=POINTER:  GOTO SETPNTR
        TOP=POINTER:    GOTO SETPNTR
RETURNX RETURN FLAG,COUNT,POINTER
```

Listing 2. FASTFIND routine in Tarbell BASIC.

GET Paid for using your Computer

FUN! Easy

RUSH COUPON FOR FREE FACTS

GREAT SPARE TIME

Send today to DAR M8

3110 Fulton Ave., Sacramento CA 95821 ✓136

CASH

NAME

STREET

CITY

STATE ZIP

FREE ADVERTISING

The Software Trader ✓229
423 Bedford Rd.
Schenectady, NY 12308

The classified magazine devoted to microcomputer software, exclusively. Let us help end your search for software. Be sure to write your ad exactly as you want it printed.

Subscription rate $7.00 per year (12 issues).

Advertising rates:

All user, noncommercial, or club advertising is free. The only requirement for free advertising is; be a subscriber.

Commercial and display advertising is also free, you pay only if you sell your product or service, at the following rates:

For each 47 character line, per issue $1.62

For each 2" wide x 3" long camera ready display ad per issue. $17.25

For each 4" wide x 3" long camera ready display ad per issue. $34.50

NOTE: Display ads will be accepted in only 2 sizes; 2"w x 3"l & 4"w x 3"l

FLOPPY SAVER

Floppy disks and programs are expensive. If you are losing programs and floppy disks due to damaged center holes, FLOPPY SAVER is the answer. FLOPPY SAVER will save disks already damaged and will protect new disks for many times normal life. FLOPPY SAVER will prevent scuffing, crimping and dimpling caused by the hub drive. FLOPPY SAVER is a 7-mil mylar reinforcing ring with a super adhesive backing. FLOPPY SAVER kit (25 rings and tool) is only $14.95 for 5¼" disk, $16.95 for 8", postpaid.

Visa / Master Charge accepted.

Dealer inquiries invited.

TRI-STAR CORPORATION ✓233
P. O. Box 1727 • Grand Junction, CO 81502
(303) 243-5200

MAILROOM PLUS©

Make Your **TRS 80** Work Like A Mini-IBM!

Mailroom Plus was developed for the National Rifle Association membership mailings. It features sorting by last name or member number in addition to zip code. The program will sort 500 names in 30-40 minutes, kill duplicates, and close up the file. **Mailroom Plus** will also search all records for category, name, state, zip (or any other search code) and print these records on labels or in tabular form. It separates large files into smaller ones by state or zip or merges small files into one large one. **Mailroom Plus** is available on 32-48K disk for $75.00 by first class mail. Order yours today postpaid.

THE PERIPHERAL PEOPLE ✓112
PO Box 524, Mercer Island, WA 98040
206-232-4505
Master Charge and VISA cards welcomed

master charge — VISA

TRS-80® DISK FILE MESS?

Find disk files instantly with the most sophisticated disk indexing program available. Similar Indexing programs exist but none with as many features as FINDISK-II!

AUTOMATICALLY create, sort, print, search a Master Index of all disk files.
AUTOMATICALLY read file names, disk numbers *(no hand entry)*.
AUTOMATICALLY print disk labels *(optional)*.
AUTOMATICALLY update Index from revised disks.
AUTOMATICALLY add optional file descriptions.
AUTOMATICALLY purge disks of unwanted files.

- **FINDISK-II** *(on tape or disk, 32K min)* $20.00

Other powerful programs from Documan available on tape or disk:

- **STRUCT-I** beam design and moment transfer w/graphics *(16K)* $15.00
- **SOLAR-I** essential calculations for passive solar design *(32K)* $30.00
- **RIA-I** complex analysis of real estate investments *(32K)* $30.00
- **DEPRECIATE-I** calculate, print 12 facts on depreciable items *(16K)* $10.00

✓145 **VISA OR MASTER CHARGE**

DOCUMAN SOFTWARE BOX 387-A KALAMAZOO, MI 49005 (616) 344-0805

TRS-80 & OTHER NEEDS FILLED FOR LESS

+ + + COMPATIBLE DISK DRIVES WITH POWER SUPPLY AND CASE—120 DAY WARRANTY + + +

- 40 TRACK (204,800 BYTE/DISK) USE BOTH SIDES, ANTI-CRIMP/POWER PROTECT **$339**
- 8 IN. DRIVE & P.S./CASE **$790** WITH P.S./CASE FOR 3 DRIVES **$950**
- 77 TRACK (197,120 BYTE) 90 DAY WARRANTY **$450**
- 4-DRIVE CABLE **$31** •• 10 DISKS-5 IN. @ **$25** -8 IN. @ **$39**-HARD CASE **$3** & **5**

••• BASE 2 PRINTER-(60 LINE/MIN, 72,80,96,120 or 132 CHAR. LINE BI-DIRECT DOT MATRIX IMPACT 96 CHAR ASCII, 15 BAUD RATES) **$499** 2K BUFFER, GRAPHICS, & TRACTOR OPTION AVAIL.•••

- CENTRONICS 737 **$840** + + + + + CABLE @ **$28**
- HARRIS SELECTRIC (WORD PROCESSING-TYPEWRITER & PRINTER) **$790**
- LOWER CASE FOR CENTRONICS 779/RADIO SHACK LINE PRINTER 1 -EASY INSTALL **$99.95**
- UPS (UNINTERRUPTIBLE POWER SUPPLY) PREVENT POWER DROP SURGE OR OUT? FROM **$195**
- CAT MODEM (ORIG/ANS) **$157** + + + + + + + + + 16K MEMORY SET (200 NANO) **$57**
- 16K L2 RADIO SHACK COMPUTER SYSTEM **$649**
- APPLE, ATARI, RADIO SHACK MODEL 1/2 HARDWARE/SOFTWARE DISCOUNTED A/R, A/P, G/L, P/R FOR **$360** or **$100** EA. (MODEL 1) & **$630** OR **$165** EA. (MODEL 2). APPLICATIONS INTERACT & ARE COMPLETE & PROFESSIONAL. WILL RUN ON OTHER COMPUTERS. THIS IS A SPECIAL INTRODUCTORY PRICE.
- ASK FOR FREE FLYER WITH OUR LOW PRICES—DEALER INQUIRIES INVITED MASS. RESIDENTS ADD 5% TAX—F.O.B. TEWKSBURY—FREIGHT EXTRA.

M/C, VISA OR CHECK ACCEPTED. TRS-80 IS A REG. TRADEMARK OF TANDY CORP.

✓140

OMNITEK SYSTEMS — 24 MARCIA JEAN DR., DEPT. K, TEWKSBURY, MA 01876 CALL 617-851-3156

ANNOUNCEMENT

The manufacturer of the FINDEX portable microcomputer with Bubble Memory seeks exclusive distributors. Financial strength and programming know-how necessary. This is not a franchise. For information contact: FINDEX, Inc., 1625 W. Olympic Blvd., #8080 Los Angeles CA USA.

FINDEX

1625 W. Olympic Blvd.
Suite 707 ✓245
Los Angeles 90015
California, U.S.A.
Phone (213) 380-6950
(213) 7-FINDEX

TRSpeed-up Revisited

With this mod, you can change speeds on the fly.

Roy A. McCoy W1JCV
Box 757
S. Lancaster, MA 01561

Have you ever wanted faster graphics or CLOAD and CSAVE operations on your TRS-80? Perhaps you've felt that it would be nice if programs, both your own and commercial ones, ran faster. For example, if you're a chess buff and own one of the Microchess programs, I'm sure you've wished you didn't have to wait so long for the computer's next move.

The circuit shown in Fig. 1 and the modifications outlined below came about as a result of my wanting to speed up the Sargon chess program. After making a few inquiries, I learned that the slowness of the program was related to the clock frequency of the TRS-80 CPU. I discussed the problem with my friend, Ed Fortmiller, who is a computer programmer. A week later he came up with the circuit shown in Fig. 1.

After deciding not to use any of the spare gates in the TRS-80, we went to the nearest Radio Shack and purchased the needed parts. During the next week I wired the circuit and made the needed modifications. The circuit worked the first time around and has been working perfectly ever since.

The total cost of the necessary parts (if purchased at Radio Shack) should be around \$6 or \$7. You will need three or four feet of small gauge hookup wire. If you don't have any available, you'll have to spend an additional \$2 for a roll of no. 22 or 24 stranded hookup wire.

The Z-80 microprocessor in the TRS-80 (known as the CPU) operates on a clock frequency of 1.774 MHz. By changing that frequency to 2.66 MHz you effect a 50 percent speed-up in the overall operation of your TRS-80.

Scott King's article ("TRSpeed-up," *Microcomputing*, September 1979, p. 138) suggests simply installing a manual switch to change the Z-80 clock frequency. However, the problem with this method is that you can't switch frequencies after power-up, because the Z-80's operation is interrupted and the TRS-80 comes to a standstill. The only way to get it going again is to turn the power off and start over.

Manual switching is a problem because the TRS-80's dynamic RAMs must be refreshed periodically by the Z-80 to remember the data stored in them. If the Z-80 is interrupted for more than one millisecond, the RAMs will not be refreshed and will forget everything they know! . . . and the TRS-80 will cease to function.

So while manual switching will get your TRS-80 running at 2.66 MHz, you won't be able to run any of your old programs or any commercial programs since they will not load at the new clock frequency. Nor can you load them at 1.774 MHz and then switch to 2.66.

The Circuit

To simplify matters, let's say that the circuit is an electronic switch. It performs the same function as the manual switch referred to earlier and even uses a manual switch.

In this case, however, S1 does not actually switch the clock frequency. It has that effect, but the actual switching is done electronically by two integrated circuits (ICs), 74LS76 and 74LS00. Although the latter IC (hereafter referred to as Z2) is shown as three separate units, it is a single IC.

As can be seen from Fig. 1, Z2 has two separate clock signals coming into it—one from pin 12 of Z56 and one from pin 8 of Z56. The position of S1, which is connected to the 74LS76 (hereafter referred to as Z1), determines which of the signals appears at pin 8 of Z2 and is passed on to Z72. The output of Z72 goes to the Z-80. Whenever S1 is operated, the switching is done so quickly by Z1/Z2 that neither the Z-80 nor the RAMs are adversely affected. That means that you can change the speed of your TRS-80 after power-up without having to worry about the RAMs' losing their data.

For example, you can load in any program and switch to the 2.66 clock frequency, and the program will run 50 percent faster. You can even change speed while a program is being executed. You can also CSAVE any program that is in the TRS-80's memory at the faster speed so that the next time you wish to CLOAD that program you can do so at the faster speed, thus taking considerably less time.

A schematic diagram of the TRS-80 is helpful (although not necessary) as you undertake the modifications to be described. A complete wiring diagram of the TRS-80, plus other useful information, can be found in the *TRS-80 Microcomputer Technical Reference Handbook*, available at Radio Shack stores.

Although the modifications to the TRS-80 and the wiring of the circuit shown in Fig. 1 are simple and easy to do, I do not recommend that a novice undertake these modifications unless

he has a more experienced friend looking over his shoulder. In writing this article, I have assumed that the readers who decide to undertake these modifications have some past experience working with electronic circuits. Thus, there are no detailed instructions for wiring the circuit. I assumed that you could do the wiring from the diagram.

The modifications described below apply to both Level I and Level II machines, 4K or 16K. You may have difficulty if your TRS-80 has a numeric keypad because it may be difficult to find a convenient place to mount the circuit shown in Fig. 1. The best thing to do is to remove the top cover of your TRS-80 and determine what space is available.

The changes described in this article, or any similar changes, will void the 90-day warranty on your TRS-80. For that reason, you may wish to wait until that period is over before modifying your unit.

Modification Steps

1. Remove from the bottom of the TRS-80 the six screws that hold the unit together, carefully labeling the holes and the screws with masking tape to make sure you return each screw to the proper hole later on. The lengths of the screws vary. After turning the unit right side up again, remove the top cover. You now have access to the foil side of the logic board where several modifications will be made.

2. Locate Z56 (74LS92). If you have a Level II machine, gaining access to the desired pins may be difficult since in some TRS-80s the piece of foam supporting the Level II hardware is glued to the board right over the pins of Z56. If so, carefully lift the foam away with a sharp knife so that you can get to the pins and the necessary trace.

3. Locate pin 8 of Z56 and the trace that leads from it to a feed-through hole that is covered with solder. Carefully cut this trace with a knife or other appropriate tool. Make sure that the trace is completely severed. Carefully remove any metal particles that may result from cutting the trace.

Now solder an eight-inch piece of hookup wire to the section of the trace going to pin 8. Solder another piece of the same length to the section of the trace going to the feed-through hole. If it's difficult to solder to the trace, you can solder the wires to pin 8 and the feed-through hole, respectively. The free ends of these wires will be connected later.

4. Locate pin 12 of Z56. Solder an eight-inch piece of hookup wire to this pin. After you've made the connection, carefully examine the nearby pins and traces for solder bridges.

5. Turn the logic board over so that you are looking at the component side. Keep the keyboard close to its spacers above the logic board so you don't put any strain on the connection between the two boards. The plastic assembly holding the power/video/tape jacks should be on your right as you look down at the board.

6. Locate Z43. (Each IC has its number printed on the board to the right of it.) The pin nearest to you on the right side of Z43 is pin 1. The next pin—on the same side, going away from you—is pin 2. Solder a 3 inch piece of hookup wire to this pin.

Now locate Z56. (It may be partly hidden by the ribbon cable from your Level II hardware.) The pin nearest to you on the left side of Z56 is pin 14. Solder the free end of the wire from pin 2 of Z43 to pin 14.

R1 4.7k, ½ Watt carbon resistor, 5 percent

R2 4.7k, ½ Watt carbon resistor, 5 percent

S1 SPDT subminiature toggle switch (RS 275-613)

Z1 74LS76 Schottky IC (RS 276-1921)

Z2 74LS00 Schottky IC (RS 276-1900)

Table 1. Parts list.

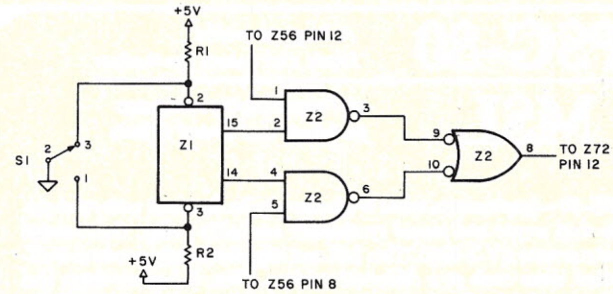


Fig. 1. Modification circuit.

7. Locate the video jack. Immediately behind it are two blue electrolytic capacitors—one mounted horizontally and the other mounted vertically. On either side of the smaller, vertically mounted capacitor are two transistors (black, molded plastic body). The one on the left is Q2.

Right behind Q2 are four small resistors. The one farthest from Q2 is R30 (47 Ohms). Connect an 8 inch piece of hookup wire to the right lead of R30. Feed the other end of this wire through the hole in the board between the video and tape jacks. This lead will later be connected to the +5 volt point that will supply voltage to Z1 and Z2.

This completes the wiring on the component side of the board. Turn the board over so that you are again looking at the foil side.

8. Locate the power jack. Just above the jack are three terminals covered with solder. Solder a six-inch piece of hookup wire to any one of them. This completes the modifications on the TRS-80 logic board.

Wiring the Small Circuit Board

The next step is wiring the circuit in Fig. 1. I did not include any step-by-step instructions for wiring this circuit. If you are unable to do this, persuade a friend to do the job for you, send me $1 for detailed step-by-step instructions or forget this article and purchase one of the TRS-80 speed-up boards advertised elsewhere in this magazine.

The circuit of Fig. 1 can be wired on an experimenter's PC board (RS 276-151) or on a small piece of IC perfboard. If you choose the latter, you can use RS 276-1394, which is much larger than necessary, but you can easily cut a 2½ X 2¾ inch piece from it. Such a piece is adequate to hold the necessary components.

Whether you use perfboard or a PC board, use sockets for the ICs. If you opt for a perfboard, be sure to purchase the sockets with long pins (RS 276-1993). By bending the pins at right angles to the perfboard, the socket can be firmly mounted on the board.

Connections must be made to other pins (not shown in Fig. 1) on both Z1 and Z2. On Z1, pin 5 is connected to +5 volts and pin 13 is grounded. Pins 1, 4, 6, 9, 12 and 16 are all soldered to a common tie point and connected through a 1k resistor (R3) to +5 volts. On Z2, pin 14 is connected to +5 volts and pin 7 is grounded.

The most convenient place to mount S1 is on the right side, top half of the TRS-80 cover, about 5 5/16 inches from the rear and about ½ inch from the lower edge of the top half of the cover. Exercise great care in drilling the mounting hole for S1. Begin with a small drill, preferably 1/16 inch, and work up gradually to the desired hole size. I used nine or ten bits to achieve the desired hole size, thus avoiding any possibility of cracking the plastic cover.

Final Steps

After you've completed the wiring of the small circuit board and drilled the mounting hole for S1, mount the board in a convenient spot on the foil side of the TRS-80 logic board. This can best be done by following the method used for securing the Level II hardware to the logic board.

Take a piece of the soft packaging material that came with your TRS-80, cut it to ap-

proximately the same size as the circuit board and then glue this piece of material to the bottom of the board. Once the glue has set, apply glue to the bottom side of the foam and press it firmly against the logic board.

Mount S1 (horizontally or vertically) in the hole drilled earlier. Tighten the mounting nut enough to hold S1 firmly in place, but do not overtighten it. Remember that the computer will be in the speeded-up mode when pin 2 of Z1 is grounded.

Solder the wire from pin 8 of Z56 to pin 5 of Z2. Cut away any excess wire. Now solder the wire from the feed-through hole to pin 8 of Z2. Cut away any excess wire. Next solder the wire from pin 12 of Z56 to pin 1 of Z2.

Solder the wire from R30 to the +5 volt point on the small circuit board. Next solder the wire from the ground terminal at the rear of the board to the ground terminal on the small circuit board.

This completes the modification of your TRS-80. Before powering up your unit, I suggest that you (or a friend) carefully review the modifications to the logic board and the wiring of the small circuit board.

Your TRS-80 will now operate at either 1.774 MHz or 2.66 MHz. To get some idea of the improvement in speed of operation, load one of your programs in at the normal speed, keeping track of the loading time. Now flip the switch to the faster speed and CSAVE the program. Next reload the program (without touching the switch), noting the loading time. Even without looking at a watch you will notice the asterisk blinking much faster.

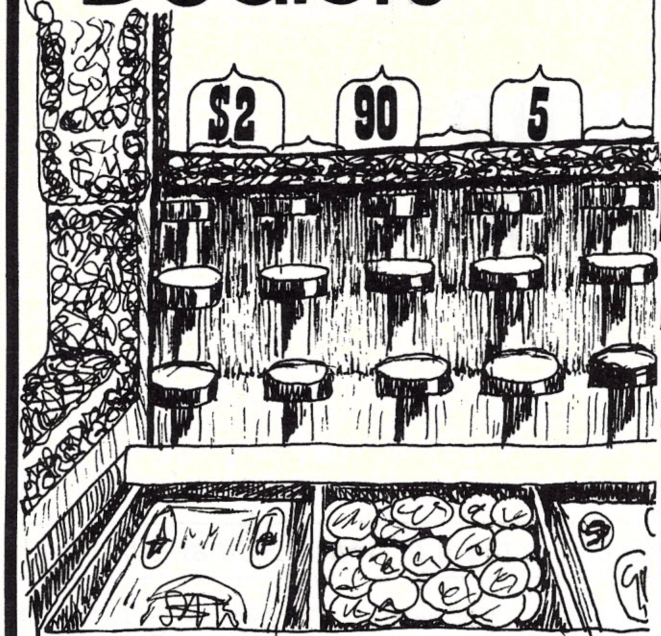
Another possibility is to enter the following two-line program and measure the time between the appearance of the words GO and STOP.

```
10 CLS: FOR J = 1 TO 200: NEXT J: PRINT
"GO"
20 FOR N = 1 TO 5500: NEXT N: PRINT
"STOP"
```

You're measuring how long it takes the computer to count from one to 5500, and you'll discover that with the Z-80 running at 2.66 MHz it counts 50 percent faster than at 1.774 MHz.■

Main/Frames from $200

Attention Dealers



Make Money... Sell

kilobaud
MICROCOMPUTING T.M.

Selling Kilobaud MICROCOMPUTING, the most complete journal of microcomputing, brings the computer enthusiast through your door. Once he's in your store, you can sell him anything.

For information on selling Kilobaud MICROCOMPUTING, call 603-924-7296 and speak with Ginnie Boudrieau, our bulk sales manager, or write to her at Kilobaud Microcomputing, 80 Pine Street, Peterborough, NH 03458.

Our dealers are telling us that Kilobaud MICROCOMPUTING is the hottest-selling computer magazine on the newsstand, so call today and join the ranks of dealers who make money with KM.

kilobaud
MICROCOMPUTING T.M.
80 Pine Street, Peterborough NH 03458

ATTENTION PROGRAMMERS:

Instant Software is looking for more good programs.

YOU may be sitting on a gold mine. The chances are good that you have developed programs for personal, hobby, business or educational use. Such programs could become best sellers in our software line and bring you a substantial monthly royalty check.

Instant Software can save you the time and expense of trying to sell your own software. Our experienced staff of topnotch programmers, marketing and advertising specialists, writers, editors and artists can package your program and make it stand out in the crowd. And . . . Instant Software's proven track record of steadily increasing sales means larger royalties for you.

Our advertising campaign will put your program before the readers of Kilobaud/Microcomputing, the Microcomputing Industry Newsletter, 80 Microcomputing, and other national publications. Our aggressive sales force has put Instant Software into more than 250 computer stores across the United States and in several foreign countries. Wherever there are microcomputers, there's Instant Software.

We are looking for good business, utility and education programs on almost any subject. Specifically, for business, we are looking for General Ledger, Cost Accounting, Accounts Receivable/Accounts Payable, Inventory, Order Entry and Payroll programs. They should be interactive, operate from a common data base, and feed into a Trial Balance Sheet and a Profit and Loss Statement. For the utilities, we'd like any system software that is useful, solves problems and will save time and/or effort for other programmers. For educational software, we want tutorial programs, educational simulations, administrator/teacher aids, mixed-media presentations, interactive learning and anything else that's creative and exciting to potential students. Those students can range from preschool age to adults; the subject matter can be anything that you know well. In short, send us software that can solve real problems for the final user.

We are presently looking for software for the following systems: TRS-80 Level II and Model 2, Apple, Heath H-8 and H-89, OSI C1P, Atari,TI 99/4, PET and CP/M.

Submitting programs is easy! Simply send us a disk containing your program or a cassette tape with your program recorded twice on each side. Be sure to include a doublespaced, typewritten explanation of (1) what the program does, (2) how to load and use it, (3) what computer it will run on, and (4) who might be interested in buying it. We'd also like a program listing as well as definitions of all variables used in the program. Include a sample data file if necessary. Then send it to:

Instant Software, Inc., Editorial Department
Peterborough, NH 03458

Our reviewers will examine your programs and (assuming we like what we see) you'll be contacted regarding the royalty arrangements.

If you want additional information about Instant Software, send us your name and address. Ask for our software documentation guide, which explains the submission procedure in greater detail.

The market for expertly written and documented software is wide open, and there is a huge demand for programs to keep up with the expanding sales of microcomputers. Get a piece of the action for yourself!

✓40 Instant Software Inc.
Peterborough, New Hampshire 03458 603-924-7296

File Dump For FLEX

Become acquainted with your disk files; FD prints them in both ASCII and hex.

Phil Hughes
PO Box 2847
Olympia WA 98507

FD is a program that prints the contents of disk files in both ASCII and hexadecimal. I originally wrote FD to determine the cause of an incompatibility between the TSC Text Editor and Ed Smith's Macro Assembler. FD is similar to a memory dump in that once you have it, you will find all kinds of reasons to use it.

Before I talk a lot about FD, let me talk a little about FLEX, which is an operating system written by Technical Systems Consultants. It was designed to run on a Southwest Technical Products 6800 computer system with either a mini-floppy or full-sized floppy disk. A new version that runs on a Smoke Signal Broadcasting system is now available. Rumors are that FLEX will also support the hard disk, which is soon to come from SWTP. FLEX does a lot for the user, and, with some additional utility programs, it can do a lot more. FD is such a utility program.

FD should be installed by saving its binary as FD.CMD on your system drive. Note that FD is loaded into the FLEX utility command area, and therefore it must be saved using the SAVE.LOW command. Once installed, it can be run by entering FD filespec, where filespec is a standard FLEX file specification, such as 1.MYFILE.TXT. The file will be printed out in hexadecimal, and all printable bit combinations will also be printed in ASCII on the right.

Listing 1 shows a sample ex-

```
+++FD TESTFILE
0000 54 68 69 73 20 69 73 20 61 20 74 65 73 74 20 66   This is a test f
0010 69 6C 65 20 74 6F 20 73 68 6F 77 20 68 6F 77 20   ile to show how 
0020 46 44 20 77 6F 72 6B 73 2E 0D 54 68 69 73 20 69   FD works.?This i
0030 73 20 61 20 74 65 78 74 20 66 69 6C 65 2C 20 74   s a text file, t
0040 68 65 72 65 66 6F 72 65 20 65 76 65 72 79 74 68   herefore everyth
0050 69 6E 67 20 69 73 20 70 72 69 6E 74 61 62 6C 65   ing is printable
0060 0D 65 78 63 65 70 74 20 74 68 65 20 63 61 72 72   ?except the carr
0070 69 61 67 65 20 72 65 74 75 72 6E 73 2E 0D 00 00   iage returns.???
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ????????????????
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ????????????????
00A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ????????????????
00B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ????????????????
00C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ????????????????
00D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ????????????????
00E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ????????????????
00F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00      ???????????????

READ PAST END OF FILE
+++LIST TESTFILE
This is a test file to show how FD works.
This is a text file, therefore everything is printable
except the carriage returns.
```

Listing 1. FD printout.

Listing 2. FD and FLEX.

```
0000                NAM    FD           - FLEX FILE DUMP
0000                OPT    XRF
              * HEX/ASCII FILE DUMP FOR FLEX 2.0
              * SSC 7-5-79 V1.4
              *:0.FLEXEQU.INC GET FLEX EQUATES
A080          F2LBUF   EQU    X'A080'   LINE BUFFER
AC00          F2BACK   EQU    X'AC00'   BACKSPACE CHARACTER
AC01          F2DEL    EQU    X'AC01'   DELETE CHARACTER
AC02          F2EOL    EQU    X'AC02'   END OF LINE CHAR.
AC03          F2DEPC   EQU    X'AC03'   PAGE DEPTH COUNT
AC04          F2WIDC   EQU    X'AC04'   WIDTH COUNT
AC05          F2NULL   EQU    X'AC05'   NULL COUNT
AC06          F2TABC   EQU    X'AC06'   TAB CHARACTER
AC07          F2DX     EQU    X'AC07'   DUPLEX MODE
AC08          F2EJC    EQU    X'AC08'   EJECT COUNT
AC09          F2PAUS   EQU    X'AC09'   PAUSE CONTROL
AC0A          F2ESC    EQU    X'AC0A'   ESCAPE CHARACTER
AC0B          F2SDRV   EQU    X'AC0B'   SYSTEM DRIVE NUMBER
AC0C          F2WDRV   EQU    X'AC0C'   WORKING DRIVE
AC0E          F2SDR    EQU    X'AC0E'   3 BYTE DATE REGISTER
AC11          F2LAST   EQU    X'AC11'   LAST TERMINATOR
```

ecution of FD. The left-most column is a relative offset address within the file. This is followed by 16 bytes of data displayed in ASCII. Any nonprintable character is replaced by a question mark (?).

Listing 2 is FD assembled from FLEX 2.0. It was assembled using RRMAC from Ed Smith's Software Works. The significant difference between the RRMAC format and Motorola assembler format is in the formation of constants. For example, the constant hexadecimal 1234 is represented by X'1234' instead of $1234. The object code generated by RRMAC is relocatable—in other words, it can be loaded anywhere in memory.

This relocation is performed by another Ed Smith program called a loader. For those who wish to use standard Motorola assembler format, the directive ORG $A100 should be added after line 134, and the constants should be changed back to the Motorola format.

Lines 6 through 133 have been included from another file. These lines are a set of equates for all the FLEX interfaces that I feel I might use.

Although most of these equates are never used in FD, it ensures that all of my programs use the same names for the FLEX interfaces. This way, if TSC releases a new version of FLEX with some of the entry points changed, I will only have to change the include file and then reassemble all my routines.

Lines 135 through 150 open the file that is to be dumped. GETBLK (line 151) is the start of the processing loop. This routine gets characters from the file and checks for a read error. DLINES (line 165) is the beginning of the routine that prints the data. Routine DONE (line 212) closes the file and returns to FLEX at its warm-start entry.

If an I/O error occurs, the routine FIL.ERR will report the error, close any open files and return to FLEX. Note that FD has no way to determine where the last piece of meaningful data is located in the file; therefore, a normal termination is the message READ PAST END OF FILE. ■

```
 22   AC12        F2UCTA    EQU   X'AC12'   USER COMMAND TABLE ADDR
                                            ESS
 23   AC14        F2LBP     EQU   X'AC14'   LINE BUFFER POINTER
 24   AC16        F2ERR     EQU   X'AC16'   ESCAPE RETURN REGISTER
 25   AC18        F2CURC    EQU   X'AC18'   CURRENT CHARACTER
 26   AC19        F2PREC    EQU   X'AC19'   PREVIOUS CHARACTER
 27   AC1A        F2LINE    EQU   X'AC1A'   CURRENT LINE NUMBER
 28   AC1B        F2LDAO    EQU   X'AC1B'   LOADER ADDRESS OFFSET
 29   AC1D        F2TRAN    EQU   X'AC1D'   TRANSFER FLAG
 30   AC1E        F2TRNA    EQU   X'AC1E'   TRANSFER ADDRESS
 31   AC20        F2ERRT    EQU   X'AC20'   ERROR TYPE
 32   AC21        F2SIOF    EQU   X'AC21'   SPCIAL I/O FLAG
 33   AC22        F2OTSW    EQU   X'AC22'   OUTPUT SWITCH
 34   AC23        F2INSW    EQU   X'AC23'   INPUT SWITCH
 35   AC24        F2FOA     EQU   X'AC24'   FILE OUTPUT ADDRESS
 36   AC26        F2FIA     EQU   X'AC26'   FILE INPUT ADDRESS
 37   AC28        F2COMF    EQU   X'AC28'   COMMAND FLAG
 38   AC29        F2CCOL    EQU   X'AC29'   CURRENT OUTPUT COLUMN
 39   AC2B        F2MEND    EQU   X'AC2B'   MEMORY END
 40   AC2D        F2ENV     EQU   X'AC2D'   ERROR NAME VECTOR
 41   AC2F        F2FIEF    EQU   X'AC2F'   FILE INPUT ECHO FLAG
 42                         *
 43                         * FLEX SUBROUTINES
 44                         *
 45   AD00        COLDS     EQU   X'AD00'   COLDSTART ENTRY
 46   AD03        WARMS     EQU   X'AD03'   WARM START ENTRY
 47   AD06        RENTER    EQU   X'AD06'   MAIN LOOP RE-ENTRY
 48   AD09        INCH      EQU   X'AD09'   INPUT CHAR
 49   AD0C        INCH2     EQU   X'AD0C'   INPUT CHAR
 50   AD0F        OUTCH     EQU   X'AD0F'   OUTPUT CHAR
 51   AD12        OUTCH2    EQU   X'AD12'   OUTPUT CHAR
 52   AD15        GETCHR    EQU   X'AD15'   PREFERRED GET CHAR
 53   AD18        PUTCHR    EQU   X'AD18'   PREFERRED PUT CHAR
 54   AD1B        INBUFF    EQU   X'AD1B'   INPUT TO LINE BUFFER
 55   AD1E        PSTRNG    EQU   X'AD1E'   PRINT STRING
 56   AD21        CLASS     EQU   X'AD21'   CLASSIFY CHARACTER
 57   AD24        PCRLF     EQU   X'AD24'   PRINT CR, LF
 58   AD27        NXTCH     EQU   X'AD27'   NEXT CHARACTER
 59   AD2A        RSTRIO    EQU   X'AD2A'   RESTORE I/O VECTORS
 60   AD2D        GETFIL    EQU   X'AD2D'   PARSE FILE SPEC.
 61   AD30        LOAD      EQU   X'AD30'   FILE LOADER
 62   AD33        SETEXT    EQU   X'AD33'   SET EXTENSION
 63   AD36        ADDBX     EQU   X'AD36'   ADD ACC-B TO X
 64   AD39        OUTDEC    EQU   X'AD39'   OUTPUT DECIMAL NUMBER
 65   AD3C        OUTHEX    EQU   X'AD3C'   OUTPUT HEX CHARACTER
 66   AD3F        RPTERR    EQU   X'AD3F'   REPORT ERROR
 67   AD42        GETHEX    EQU   X'AD42'   GET HEX NUMBER
 68   AD45        OUTADR    EQU   X'AD45'   OUTPUT HEX ADDRESS
 69   AD48        INDEC     EQU   X'AD48'   INPUT DECIMAL NUMBER
 70   AD4B        DOCMND    EQU   X'AD4B'   CALL DOS
 71   B400        FMINIT    EQU   X'B400'   FMS INITIALIZATION
 72   B403        FMSCLS    EQU   X'B403'   FMS CLOSE
 73   B406        FMS       EQU   X'B406'   FMS CALL
 74                         *
 75                         * FMS COMMANDS
 76                         *
 77   0000        FL2IO     EQU   X'0'      READ/WRITE NEXT BYTE
 78   0001        FM2OPR    EQU   X'1'      OPEN FOR READ
 79   0002        FM2OPW    EQU   X'2'      OPEN FOR WRITE
 80   0003        FM2OPU    EQU   X'3'      OPEN FOR UPDATE
 81   0004        FM2CLF    EQU   X'4'      CLOSE FILE
 82   0005        FM2RWF    EQU   X'5'      REWIND FILE
 83   0006        FM2OPD    EQU   X'6'      OPEN DIRECTORY
 84   0007        FM2GIR    EQU   X'7'      GET INFORMATION RECORD
 85   0008        FM2PIR    EQU   X'8'      PUT INFORMATION RECORD
 86   0009        FM2RSS    EQU   X'9'      READ SINGLE SECTOR
 87   000A        FM2WSS    EQU   X'A'      WRITE SINGLE SECTOR
 88   000C        FM2DLF    EQU   X'C'      DELETE FILE
 89   000D        FM2RNF    EQU   X'D'      RENAME FILE
 90   000F        FM2NSS    EQU   X'F'      NEXT SEQUENTIAL SECTOR
 91   0010        FM2OSI    EQU   X'10'     OPEN SYSTEM INFORMATION
                                             RECORD
 92   0011        FM2GRB    EQU   X'11'     GET RANDOM BYTE FROM SE
                                            CTOR
 93   0012        FM2PRB    EQU   X'12'     PUT RANDOM BYTE IN SECT
                                            OR
 94   0014        FM2FND    EQU   X'14'     FIND NEXT DRIVE
 95   0015        FM2POS    EQU   X'15'     POSITION BY RECORD
 96   0016        FM2BOR    EQU   X'16'     BACKUP 1 RECORD
 97                         *
 98                         * FILE CONTROL BLOCK SEPCIFICATIONS
 99                         *
100   0000        FB2FNC    EQU   0         FMS COMMAND
101   0001        FB2ESB    EQU   1         ERROR STATUS
102   0002        FB2ACT    EQU   2         ACTIVITY STATUS
103   0003        FB2DRV    EQU   3         DRIVE NUMBER
104   0004        FB2NAM    EQU   4         FILE NAME (8 BYTES)
105   000C        FB2EXT    EQU   12        EXTENSION (3 BYTES)
106   000F        FB2FAT    EQU   15        FILE ATTRIBUTES
107   0011        FB2SDA    EQU   17        STARTING DISK ADDRESS
108   0013        FB2EDA    EQU   19        ENDING DISK ADDRESS
109   0015        FB2SIZ    EQU   21        FILE SIZE
110   0017        FB2SMI    EQU   23        FILE SECTOR MAP INDICAT
                                            OR
111   001C        FB2FLP    EQU   28        FCB LIST POINTER
112   001E        FB2CUR    EQU   30        CURRENT POSITION
113   0022        FB2INX    EQU   34        DATA INDEX
114   0023        FB2RDX    EQU   35        RANDOM INDEX
115   0024        FB2NWB    EQU   36        NAME WORK BUFFER
116   002F        FB2CDA    EQU   47        CURRENT DIRECTORY ADDRE
                                            SS
117   0032        FB2FDP    EQU   50        FIRST DELETED DIRECTORY
                                             POINTER
118   003B        FB2SCF    EQU   59        SPACE COMPRESSION FLAG
119   0040        FB2BUF    EQU   64        SECTOR BUFFER
120                         *
121                         * FILE EXTENSIONS
122   0000        FEXT.BIN  EQU   0
123   0001        FEXT.TXT  EQU   1
124   0002        FEXT.CMD  EQU   2
125   0003        FEXT.BAS  EQU   3
126   0004        FEXT.SYS  EQU   4
127   0005        FEXT.BAK  EQU   5
128   0006        FEXT.SCR  EQU   6
129   0007        FEXT.DAT  EQU   7
130   0008        FEXT.BAC  EQU   8
131   0009        FEXT.DIR  EQU   9
132   000A        FEXT.PRT  EQU   10
133   000B        FEXT.OUT  EQU   11
```

```
                              *
     3R 0000 CE 0000          START     LDX   #0          CLEAR FILE OFFSET ADDRE
                                                          SS
     9  0003 FF 00C9                    STX   POSDMP
     15 0006 7F 00C8                    CLR   RESCNT      Clear residual count
     18 0009 CE 00DD                    LDX   #FCB
     27 * 000C BD AD2D                  JSR   GETFIL      Get file name
     31 000F 25 0E (001F)               BCS   FILERR      Error
     2R 0011 86 01                      LDA A #FEXT.TXT   Set TXT extension if
     11 * 0013 BD AD33                  JSR   SETEXT      defaulted
     13 0016 86 01                      LDA A #FM2OPR     Open for
     19 0018 A7 00                      STA A FB2FNC,X    read
     28 * 001A BD B406                  JSR   FMS         Read it
     32 001D 27 03 (0022)               BEQ   OVER
     3R 001F 7E 00BC          FILERR    JMP   FIL.ERR     File error
     2R 0022 86 FF            OVER      LDA A #X'FF'
     8  0024 A7 3B                      STA A FB2SCF,X    Turn off space compress
                                                          ion
     17 * 0026 BD AD24                  JSR   PCRLF
        0029                  GETBLK    EQU   *           GET NEXT BLOCK TO PRINT

     3R 0029 CE 00CD                    LDX   #BUFF       Buffer address
     5  002C C6 10                      LDA B #16         Bytes/line
        002E                  GETNXT    EQU   *           Get next character
     6R 002E FF 00CB                    STX   BUFPTR      Save buffer ptr
     9  0031 CE 00DD                    LDX   #FCB
     18 * 0034 BD B406                  JSR   FMS
     22 0037 27 03 (003C)               BEQ   GOTONE      No error
     5R 0039 F7 00C8                    STA B RESCNT      Save residual count
     5R 003C FE 00CB          GOTONE    LDX   BUFPTR
     11 003F A7 00                      STA A 0,X         Save character
     15 0041 08                         INX
     17 0042 5A                         DEC B
     21 0043 26 E9 (002E)               BNE   GETNXT      Loop until we have 16
     3R 0045 CE 00C9          DLINES    LDX   #POSDMP
     12 * 0048 BD AD45                  JSR   OUTADR      Output offset
     17 004B FE 00C9                    LDX   POSDMP      Bump X by X'10'
     19 004E C6 10                      LDA B #X'10'
     28 * 0050 BD AD36                  JSR   ADDBX
     34 0053 FF 00C9                    STX   POSDMP
     36 0056 86 20                      LDA A #X'20'      Space
     45 * 0058 BD AD18                  JSR   PUTCHR
     48 005B CE 00CD                    LDX   #BUFF       Buffer location
     50 005E C6 10                      LDA B #16         Count
     54 0060 F0 00C8                    SUB B RESCNT      - Residual count
     58 0063 27 48 (00AD)               BEQ   DONE        No data left
     9R* 0065 BD AD3C         XLOOP     JSR   OUTHEX
     11 0068 86 20                      LDA A #X'20'
     20 * 006A BD AD18                  JSR   PUTCHR
     24 006D 08                         INX
     26 006E 5A                         DEC B
     30 006F 26 F4 (0065)               BNE   XLOOP
     2R 0071 86 20                      LDA A #X'20'
     11 * 0073 BD AD18                  JSR   PUTCHR
     15 0076 B6 00C8                    LDA A RESCNT
     17 0079 16                         TAB
     19 007A 49                         ROL A
     21 007B 1B                         ABA
     23 007C 16                         TAB               *3
     2R 007D 86 20            ALIGN     LDA A #C' '       Space
     11 * 007F BD AD18                  JSR   PUTCHR      Space out to ASCII fiel
                                                          d
     13 0082 5A                         DEC B
     17 0083 26 F8 (007D)               BNE   ALIGN
     3R 0085 CE 00CD                    LDX   #BUFF
     5  0088 C6 10                      LDA B #16
     9  008A F0 00C8                    SUB B RESCNT
     5R 008D A6 00            CHLOOP    LDA A 0,X         Get char and check for
                                                          valid ASCII
     7  008F 84 7F                      AND A #X'7F'
     9  0091 81 7F                      CMP A #X'7F'
     13 0093 27 04 (0099)               BEQ   QMARK
     2R 0095 81 1F                      CMP A #X'1F'
     6  0097 22 02 (009B)               BHI   GOODH
     2R 0099 86 3F            QMARK     LDA A #C'?'       Substitute a ?
     9R* 009B BD AD18         GOODH     JSR   PUTCHR      Print it
     13 009E 08                         INX
     15 009F 5A                         DEC B
     19 00A0 26 EB (008D)               BNE   CHLOOP
     9R* 00A2 BD AD24                   JSR   PCRLF
     15 00A5 7D 00C8                    TST   RESCNT      All done?
     19 00A8 26 12 (00BC)               BNE   FIL.ERR     Yes
     3R 00AA 7E 0029                    JMP   GETBLK      Get next block
        00AD                  DONE      EQU   *           ALL DONE
     2R 00AD 86 04                      LDA A #FM2CLF     Close file
     5  00AF CE 00DD                    LDX   #FCB
     11 00B2 A7 00                      STA A FB2FNC,X
     20 * 00B4 BD B406                  JSR   FMS
     24 00B7 25 03 (00BC)               BCS   FIL.ERR     Error
     3R 00B9 7E AD03                    JMP   WARMS       Return to FLEX
        00BC                  FIL.ERR   EQU   *           FILE OPEN OR CLOSE ERRO
                                                          R
     3R 00BC CE 00DD                    LDX   #FCB
     12 * 00BF BD AD3F                  JSR   RPTERR
     21 * 00C2 BD B403                  JSR   FMSCLS      Close everything
     24 00C5 7E AD03                    JMP   WARMS       Return to FLEX
                              *
        00C8                  RESCNT    RMB   1           Residual count
        00C9                  POSDMP    RMB   2           Relative offset
        00CB                  BUFPTR    RMB   2           Buffer pointer
        00CD                  BUFF      RMB   16          Data buffer
        00DD                  FCB       ENT
        00DD                            RMB   320
        021D                  LAST      ENT
        021D                            END   START
 AD36 ADDBX      63   169
 007D ALIGN     190   193
 00CD BUFF      228   152   173   194
 00CB BUFPTR    227   155   160
 008D CHLOOP    197   207
 AD21 CLASS      56
 AD00 COLDS      45
 0045 DLINES    165
 AD4B DOCMND     70
 00AD DONE      212   176
 AC00 F2BACK      7
 AC29 F2CCOL     38
 AC28 F2COMF     37
 AC18 F2CURC     25
 AC01 F2DEL       8
 AC03 F2DEPC     10
 AC07 F2DX       14
 AC08 F2EJC      15
 AC2D F2ENV      40
 AC02 F2EOL       9
 AC16 F2ERR      24
 AC20 F2ERRT     31
 AC0A F2ESC      17
 AC26 F2FIA      36
 AC2F F2FIEF     41
 AC24 F2FOA      35
 AC23 F2INSW     34
 AC11 F2LAST     21
 AC14 F2LBP      23
 A080 F2LBUF      6
 AC1B F2LDAO     28
 AC1A F2LINE     27
 AC2B F2MEND     39
 AC05 F2NULL     12
 AC22 F2OTSW     33
 AC09 F2PAUS     16
 AC19 F2PREC     26
 AC0E F2SDR      20
 AC0B F2SDRV     18
 AC21 F2SIOF     32
 AC06 F2TABC     13
 AC1D F2TRAN     29
 AC1E F2TRNA     30
 AC12 F2UCTA     22
 AC0C F2WDRV     19
 AC04 F2WIDC     11
 0002 FB2ACT    102
 0040 FB2BUF    119
 002F FB2CDA    116
 001E FB2CUR    112
 0003 FB2DRV    103
 0013 FB2EDA    108
 0001 FB2ESB    101
 000C FB2EXT    105
 000F FB2FAT    106
 0032 FB2FDP    117
 001C FB2FLP    111
 0000 FB2FNC    100   144   215
 0022 FB2INX    113
 0004 FB2NAM    104
 0024 FB2NWB    115
 0023 FB2RDX    114
 003B FB2SCF    118   149
 0011 FB2SDA    107
 0015 FB2SIZ    109
 0017 FB2SMI    110
 00DD FCB       229   138   156   214   220
```

```
0008 FEXT.BAC   130
0005 FEXT.BAK   127
0003 FEXT.BAS   125
0000 FEXT.BIN   122
0002 FEXT.CMD   124
0007 FEXT.DAT   129
0009 FEXT.DIR   131
000B FEXT.OUT   133
000A FEXT.PRT   132
0006 FEXT.SCR   128
0004 FEXT.SYS   126
0001 FEXT.TXT   123   141
00BC FIL.ERR    219   147   210   217
001F FILERR     147   140
0000 FL2ID       77
0016 FM2BOR      96
0004 FM2CLF      81   213
000C FM2DLF      88
0014 FM2FND      94
0007 FM2GIR      84
0011 FM2GRB      92
000F FM2NSS      90
0006 FM2OPD      83
0001 FM2OPR      78   143
0003 FM2OPU      80
0002 FM2OPW      79
0010 FM2OSI      91
0008 FM2PIR      85
0015 FM2POS      95
0012 FM2PRB      93
000D FM2RNF      89
0009 FM2RSS      86
0005 FM2RWF      82
000A FM2WSS      87
B400 FMINIT      71
B406 FMS         73   145   157   216
```

```
B403 FMSCLS    72   222
0029 GETBLK   151   211
AD15 GETCHR    52
AD2D GETFIL    60   139
AD42 GETHEX    67
002E GETNXT   154   164
009B GOODH    204   202
003C GOTONE   160   158
AD1B INBUFF    54
AD09 INCH      48
AD0C INCH2     49
AD48 INDEC     69
021D LAST     231
AD30 LOAD      61
AD27 NXTCH     58
AD45 OUTADR    68   166
AD0F OUTCH     50
AD12 OUTCH2    51
AD39 OUTDEC    64
AD3C OUTHEX    65   177
0022 OVER     148   146
AD24 PCRLF     57   150   208
00C9 POSDMP   226   136   165   167   170
AD1E PSTRNG    55
AD18 PUTCHR    53   172   179   184   191   204
0099 QMARK    203   200
AD06 RENTER    47
00C8 RESCNT   225   137   159   175   185   196   209
AD3F RPTERR    66   221
AD2A RSTRIO    59
AD33 SETEXT    62   142
0000 START    135   232
AD03 WARMS     46   218   223
0065 XLOOP    177   182
```

Breakout Box

Accessing an extra parallel I/O port is easy with this simple hardware project.

Don Walters
2849 Verle St.
Ann Arbor, MI 48104

It was inevitable that microcomputers would find their way into the ham shack, as the two hobbies overlap. Although most of the microcomputers will perform dedicated functions such as CW, RTTY, antenna aiming for satellite work and slow-scan TV, a few will end up with an extra parallel I/O port or two. The spare I/O port will tempt its owner to experiment with bit pushing and pulling in many projects.

Every new technology brings new needs and problems with it. Such is the case with the microcomputer and its spare parallel I/O port. The problem is how are you going to get at the bits and signals on it. Just run wires out near where you need them, right?

Photo 1 demonstrates that this method works. However, a few weeks of working with it this way will quickly change your mind about how nice the method really isn't. It won't be long before you decide that there must be a better way to access the bits and signals of the parallel I/O port.

A Better Way

While watching a potential vendor use a breakout box to make his printer work with our minicomputer-based word-processing system, I realized that this technique could be applied to a parallel I/O port as well. This would allow quick accessing of the bits and signals of the parallel I/O port of my Imsai 8080.

I thought about how I could set up such a convenience and worked the whole thing out. By the end of the evening I had put the parallel I/O breakout panel together, and had it debugged.

Photo 2 shows the result of my efforts. Also note in the photograph the use of a standard connector plug for connecting the breakout panel to the parallel I/O port connectors on the back of my Imsai 8080. The constant use of standard connectors on my system has saved me many headaches over the past couple years.

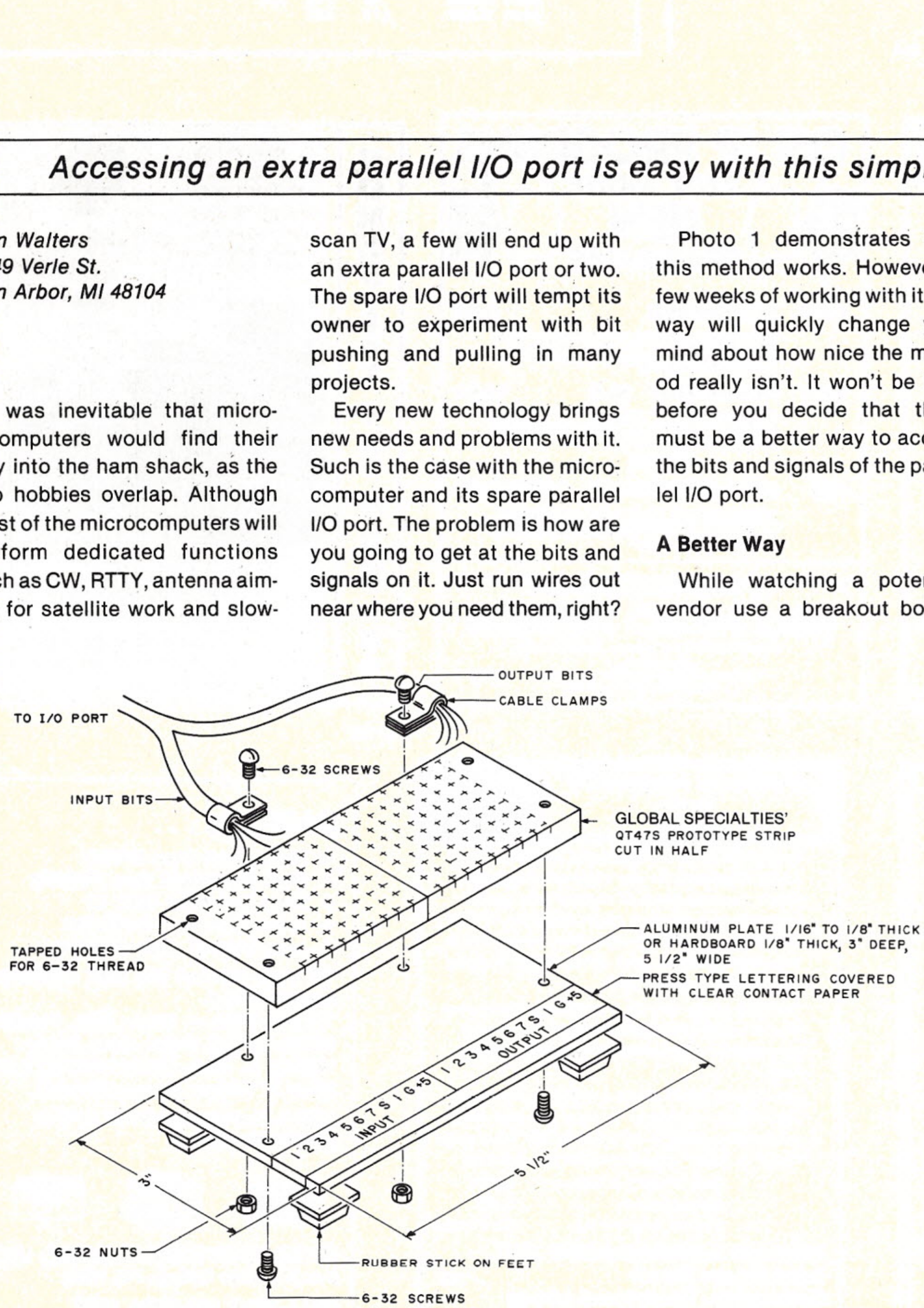


Fig. 1. General construction details for the breakout panel.

Construction

The breakout panel is simple to build and should take about an evening to put together. Your total cost will be about $15. Fig. 1 gives the general construction details on how to build the breakout panel.

The first step is to cut a piece of aluminum or hardboard to the size you want your breakout panel to be. If you want to duplicate my breakout panel use the dimensions given in Fig. 1.

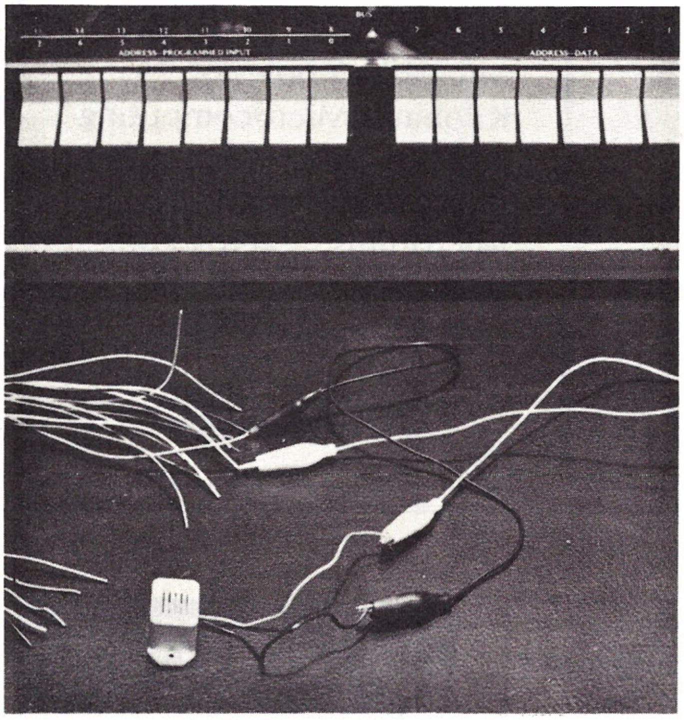


Photo 1. Cumbersome array of loose wires coming from a parallel I/O port with a solid-state buzzer precariously attached to one of the output bits (I think).

The hardest step is to cut the Global Specialties' QT47S or equivalent prototype strip (available from several of the mail-order houses that advertise in *Microcomputing*, as well as from Radio Shack and possibly local electronics outlets in your area) in two, lengthwise. You want to separate the two halves of the prototype strip so that you end up with two half-strips.

The prototype strip is made of soft plastic and can be cut with a heavy knife. Exercise care in cutting and make several passes over the same cut before you cut through the prototype strip's center. Alternately, you can use a small hobby-type hacksaw to cut the protostrip in two.

Lay half of the prototype strip and the cable clamps on the base and mark where you will have to drill their mounting holes. Use presstype lettering to label the holes of the prototype strip as to which bit or signal is available at that point on the prototype strip. Photo 3 shows how I labeled the various holes on the prototype strip. Cover the lettering with a strip of clear contact paper to protect it.

Assemble all the parts onto the base using six to 32 machine screws and nuts. The easy way to mount the prototype strip is to tap the holes on either end with a 6-32 tap. Remember to leave the cable clamps a little loose, since they will not be tightened until all the wires have been attached to the prototype strip.

After all the components are mounted to the base, attach the rubber feet.

The last step is to attach the wires coming from the parallel I/O port to the modified prototype strip. Number 24 stranded wire is about the right size. Cut about a quarter of an inch of insulation from the end of each wire, then twist the wire tightly together. Tin each wire with a little solder and insert each wire into its hole in the prototype strip.

Tighten the cable clamp screws so they provide strain relief for the wire bundle going to the parallel I/O port. Make sure the various wires, attached to the prototype strip and labeled as to their bit or signal, actually carry that bit or signal from the parallel I/O port.

That's all there is to building this simple project. ■

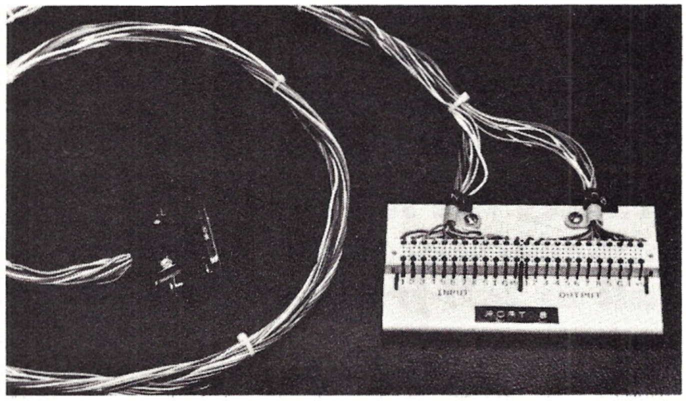


Photo 2. Breakout panel that allows easy access and connection to the various signals of the parallel I/O port. The labeling of the signals on the panel speeds up connection of projects or experiments, since I don't have to look up what each wire is. Also, you should note the connector used at the other end, which makes the breakout panel more useful since it can be used to check out any other parallel port or, for that matter, a serial port. In my case, I standardized on the DB-25 series of connectors for both RS-232 and parallel I/O. That decision three years ago has saved me many headaches ever since.

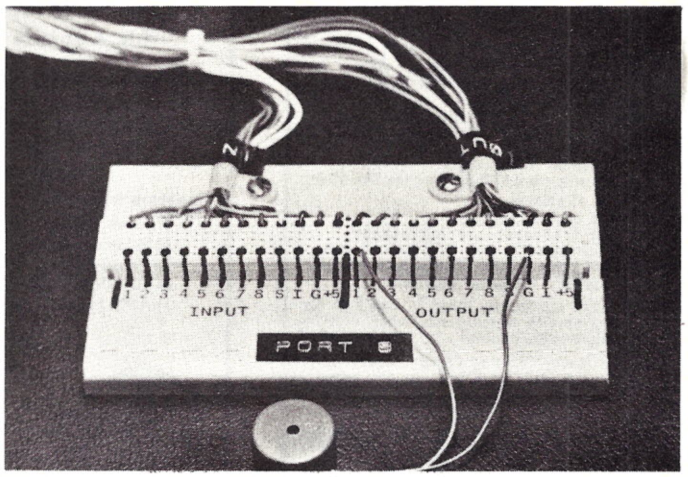


Photo 3. A close-up look at how easy it is to hook into single bits of an I/O port. Note how cleaner and easier it is to hook up a solid-state buzzer in this photo than it was in Photo 1.

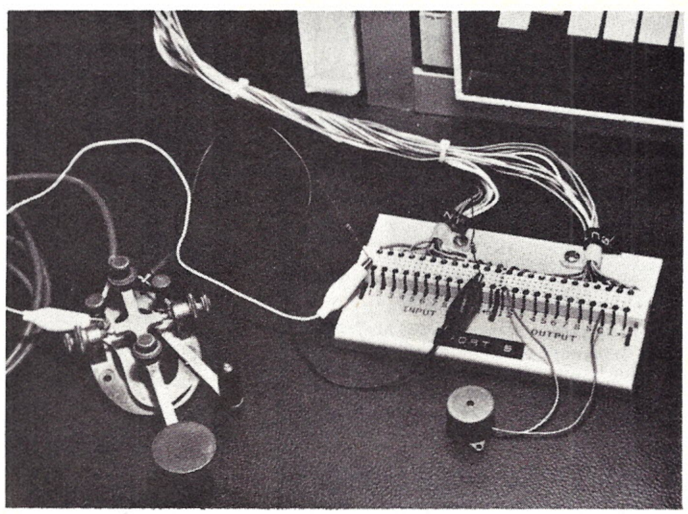


Photo 4. Close-up view of an expensive code practice oscillator? Well, not really. I was using this setup to debug a simple CW to CRT display program running on my Imsai 8080 in the background.

BASIC Programming Tips

Don't sacrifice efficiency for faster program development.

Alfred E. Williams
1551 Oceanaire Dr.
San Luis Obispo, CA 93401

Most microcomputer BASIC programmers eventually run into applications that require faster execution speed or less memory to be practical. The trend in the computer industry is to favor faster development of more reliable and maintainable programs at the expense of efficiency, except in time-critical applications. With a little thought, the programmer can develop some techniques to promote execution efficiency without affecting the speed or clarity of his programming.

The goals of reducing required memory and increasing execution speed often conflict. For example, the simple bubble sort needs minimum storage in addition to the input-output array, but is very time-consuming. More sophisticated sorts (e.g., Shell sort) execute many times faster but require more storage. I'll focus on tips to improve execution speed.

There are many different implementations of BASIC—integer or floating-point, compiler or interpreter—with varying degrees of efficiency. If you're lucky enough to have a choice of software for your micro, shop around. If efficiency is critical for you, you may be better off investing in a commercial, enhanced BASIC interpreter or compiler rather than the one supplied by your machine's manufacturer. Remember that compilers almost always have a speed advantage over interpreters in actual execution. The requirement for a compile phase may be frustrating, however, if you're used to freely twiddling with your program during debugging.

Consider your choice of storage devices for your program and data files. Generally, *direct access devices* (floppy and hard disks) provide faster retrieval of data than the various kinds of tape devices, particularly when records are to be fetched from random locations in the file rather than in the sequence they were loaded. Unfortunately, the average cost for both disks and drives is higher than the cost for tapes and tape devices, although the appearance of some new, less expensive disk devices recently has provided some hope of an affordable disk for all micro users.

Watch for Critical Inner Loops

The majority of execution time in non-input/output (I/O) bound programs is confined to less than 5 percent of the source text.[1] Donald Knuth, in his writings, points out that program execution speed may largely depend on the speed of one or more short loops. He calls these *critical inner loops.*[2]

Look for these critical loops in your program, and make sure that all statements in the loop are executed there. For example, if the value of a variable is not changed in a loop, initializing it there will waste execution time since it will be performed several times instead of once. Execution time may also be saved by doubling up the calculations in a loop. Assume you want to initialize a 40-element array to zero. One way to do this might be:

```
FOR I = 1 TO 40: A(I) = 0: NEXT I
```

Almost half the overhead of the FOR loop, however, could be saved (at a minor expense in memory) by writing the initialization this way:

```
FOR I = 1 TO 20: A(I) = 0: A(I + 20) = 0: NEXT I
```

Sometimes it is beneficial to remove the loop entirely and process using straight-line logic. If your array was four instead of 40 elements, it might have been better to initialize each element of the array individually, rather than use a FOR loop. You must evaluate the trade-off between execution speed, memory and programming ease.

All of the techniques discussed below are especially important in critical inner loops. If you repeat calculations unnecessarily, the loop will multiply its importance many times over a similar error outside the loop.

Repeating Calculations Unnecessarily

Execution efficiency can be improved by setting a variable to the result of a calculation and using that variable rather than repeating the calculation for later testing and computation. Consider the following statement:

```
IF A*B/C - D>0 THEN E = A*B/C + 1
```

Notice that A and B must be multiplied and divided by C twice. This could be avoided by replacing the statement by the following:

```
Z = A*B/C: IF Z - D>0 THEN E = Z + 1
```

Once again, the trade-off is between execution speed, memory and program clarity. Notice that the memory saved in the source text may exceed the memory required to store the new variable if the calculation is complex, depending on the amount of compression your BASIC does in storing the program. You sacrifice program clarity because you use a variable to store an intermediate result, which may make the meaning of the calculation harder to see. Because BASIC allows only one-letter variable names, using a variable this way may also be a luxury you can't afford, but keep the tip in mind for your less complex programs.

Avoid Unnecessary Subscript Calculations

If your BASIC has array variables, you have a technique that, along with FOR loops, allows you to conveniently store and manipulate variable values. Remember, however, that BASIC must spend some time evaluating the subscript and locating the position it points to in the array.

This time is multiplied with multi-subscripted variables. For this reason, it's more efficient to move the value from the subscripted variable to a new unsubscripted variable if the same location in an array is used more than once. Example 1 shows what I recently found in one of my programs. In this case, I asked BASIC to evaluate two

X1 = CO(A,1) − 30: X2 = CO(A,1) − 5: X3 = CO(A,1) + 5: X4 = CO(A,1) + 30

Example 1.

X1 = CO(A,1) − 30: X2 = X1 + 25: X3 = X1 + 35: X4 = X1 + 60

Example 2.

subscripts four times instead of once. Since this occurred in a time-critical inner loop, I saved some time by changing it to Example 2.

Avoid Unnecessary Data Conversions

If your BASIC supports more than one data format, you should find out how these formats are used by BASIC for testing and computation and avoid requiring the computer to convert between them unnecessarily. For example, if fixed-binary numbers are converted to floating point before being tested against, or used in, arithmetic operations with other floating-point numbers, you may want to move the value of a fixed-point variable to a floating-point variable used instead in expressions with other floating-point numbers. Otherwise, your BASIC has to convert the fixed-binary number to floating point every time it occurs in a test or computation with floating-point numbers.

Avoid Unnecessary Transfers of Control

There are many legitimate reasons for using GOSUB in a program; the two most important are for clarity and to use a common section of code without duplication from several different places in the program. Using the GOTO is more controversial, particularly among structured programming advocates. In any case, transfers of control do require CPU time, and you should avoid bouncing around in your program unless it's necessary.

The *top-down development* approach of structured programming uses subroutines to allow the major functions of the program to be developed first. This leaves development of the detailed processing logic for each functional program block for after the structure and relationships among the blocks are designed. If you use this approach, sometimes you'll find that some subroutines are called from only one location, and including them in the mainline of the calling block won't seriously affect its clarity. In these cases you should consider eliminating the subroutine and putting its code back into the mainline logic of the caller.

If you're not convinced about structured programming, you're even more likely to have unnecessary GOTOs between sections of your program. (The worst case—lots of inexperienced programmers make this mistake—is:

```
100 GOTO 200
200 PRINT "DUMMY, YOU WENT TO THE
STATEMENT IMMEDIATELY FOLLOWING
THE GOTO"   )
```

As you write each GOTO, think whether the routine you're jumping to could be sequentially included instead of inserting the GOTO. If so, do it!

Consider Assembler Subroutines

Finally, if your computer not only has BASIC but also an Assembler, or if you are skilled in machine language and your BASIC has an interface, you should consider functions done repeatedly, such as those in critical inner loops, for conversion to Assembler. If properly written, Assembler routines will usually outperform both the BASIC interpreters and compilers. The problem is that Assembler routines are harder to write, debug and maintain than similar routines in BASIC. For this reason, unless you're a computer freak (I am) or a dedicated masochist, I recommend you consider Assembler primarily for repetitive time-critical functions only.

It may seem that the execution time saved by each of these techniques will give a small advantage compared to the extra effort required to write your program. I certainly don't want to suggest that you significantly add to your development time in pursuing small efficiencies. Even so, I think you'll find that if you keep the principles in this article in mind, they will eventually become automatic for you; the overall time savings per program execution could be significant. ■

References

1. Dan Ingalls, "The Execution Time Profile as a Programming Tool," *Compiler Optimization,* Second Courant Computer Science Symposium, Prentice-Hall, Englewood Cliffs, NJ, 1972.

2. Donald E. Knuth, "Structured Programming with GOTO Statements," Stanford University Computer Science Department Report STAN-CS-74-416, May 1974.

Program Patching For I/O Flexibility

The author describes a technique for enhancing poorly documented software.

Ken Barbier
Borrego Engineering
PO Box 1253
Borrego Springs, CA 92004

Hey! Did you hear the one about the computer nut who went out and bought an old 5-level TTY and built his own interface for it so he could have nice hard copies of his programs? And then he found out that the supplier of his favorite version of BASIC would not disclose the source listing of the program so that our hobbyist friend could make the changes necessary to use his new printer! Well, he shouldn't be feeling lonely. There are a lot of us who have been in a similar situation, wanting to make enhancements in undocumented software.

There *is* a way to add all sorts of nice things to the programs we have purchased from tight-lipped vendors, if only we know the locations in the programs that are used to control the input and output devices. Fortunately, these locations are usually not kept secret. Even if they were, there are ways to implement hardware traps to learn the location of the hardware-controlling routines. But the subject of hardware traps is beyond the scope of this article.

Start Simple

Let us consider the simpler problem first. Suppose we want to add a hard-copy device to a BASIC-language system consisting of a CPU, a TV display terminal and an audio cassette tape drive. We have been using a BASIC compiler that allows us to save and load programs through the audio tape interface. But there is no provision for transferring the output from the TV terminal to the hard-copy device whenever we want to, so that we can save either program listings or the results of computations.

Of course, if we know the locations in BASIC that handle the input and output operations, we can always change the I/O port assignment and status word bit-pattern mask by flipping front-panel switches (if any exist on our system!) every time we want to switch from our nice quiet TV display to a noisy printer, but this is frustratingly time consuming and error prone.

Let us instead incorporate a way to switch our output from the TV display to the hard-copy printer and back with a single console keystroke, which we will make invisible to the main program. This method requires only that we know the location of the console input and output driver routines.

We will also need a little memory not accessible to the main program. This can be RAM that is not contiguous with the main memory (so BASIC doesn't know it's there), or we can reserve a little space at the top of main memory when BASIC asks "MEMORY SIZE?" (If the program you want to modify does not incorporate this feature, use "hidden" RAM, which is separated from the main storage.)

With the method described below, we can reassign any number of I/O devices at will and can even assign blocks of RAM memory as "mass" storage, making multi-pass assemblers run faster than they would in a disk-oriented system! . . . with no tape to rewind between passes.

Patching the Main Program

Somewhere in our main program is a console input routine that looks something like that shown in Example 1, or would, if we had the source listing. Since we don't, we will have to hand disassemble the code around our input routine to figure out the exact conventions used for reading the keyboard status, reading the keyboard data and passing it back to the calling program. Whatever the case, do not execute the code at this location. Instead, substitute a JUMP instruction to our input patch.

In a stack-oriented microprocessor this will leave the main program's calling address on the top of the stack. For register-oriented micros or minis we will preserve the contents of all registers anyway, so the return convention will not be disturbed.

Our input patch will look for an input character just as our main program did. But now we will not simply return to the calling program. We are going to examine the character to see if it is one of a number of special control characters we are using to implement our device reassignments. If it is, we will execute the reassignment and

jump back to the input routine to fetch the next operator keystroke. This next keystroke is the one we will send back to the main program. This way, our special characters are never seen by the calling program.

Selecting Special Characters

Any keyboard character that is *never* used by the main program qualifies as a special control character. These usually include a number of control characters, usually designated by "↑T," for example, produced by holding down the CTRL key and pressing a letter key. There are only a few control keys used by most programs, leaving us a rather large number to use for our special controls.

For our simple example we will need only two special characters. So we choose "↑T" to switch all output to the hardcopy device and "↑V" to switch all output back to the TV display. To see how these controls can be used, consider a typical BASIC programming session.

Using the Controls

BASIC is loaded from the cassette tape, and we have keyed in a program that is designed to produce some nicely formatted numeric output (I plan to use this as "Table 1" in a future article for *Kilobaud*). We try running the program, and the output is garbage. Listing the program on the TV screen 16 lines at a time does not disclose the source of the error, so we need a hard-copy listing to permit examining the program as a whole. We enter the following keystrokes: LIST ↑T CR, and our program is printed out on the TTY—from the CR we entered last through the "OK" that follows the listing. We now key in "↑V" and we are back with the TV as the output device.

Having the entire program to examine at once, we are able to spot the error that was entered while the dog was chewing on our shoelaces. Exiling the mutt from the computer shack, we correct the error, and using the TV display we run the program. This time the output looks good.

So now we roll the old yellow paper out of the TTY, roll in a sheet of nice bond paper, and key in: RUN ↑T CR, and our Table 1 appears on the TTY, nicely typed out. Don't forget to incorporate enough line feeds following the output to prevent the "OK" from appearing after the printout.

Implementing the Controls

Find the console input read routine in the main program, as discussed above, and patch in a jump to CNTIN, which is our control input routine. Example 2 is written using Intel 8080 Assembly language, but should be easily convertible to other small machines. CNTIN uses the "↑T" and "↑V" character codes to toggle a flag. CNTOU is our controlled output routine, which tests the flag to see whether the output goes to the TV display (CRTOU) or the hardcopy device (TTYOU).

The main program will also have to be patched at its output routine with a JUMP to CNTOU. CRTOU and TTYOU are device-specific driver routines for the TV display and the TTY. If the change made to the system is the addition of the TTY as decribed above, CRTOU will be the same instruction sequence as the original output routine.

Note that in Example 2 CNTOU expects that the character to be output will have been pushed onto the stack on top of the return address. This convention is used by Altair BASIC and is included here as an example of how to handle different conventions. Be sure to use the correct convention for the program you are patching.

The device driver routines themselves are hardware dependent and will vary from one machine to another and from one I/O board to another. The examples given here, therefore, do not include the actual I/O port addresses or status bit mask values.

```
INPUT   IN      STATUS   LOOK FOR OPERATOR INPUT
        ANI     MASK     MASK OFF DATA READY BIT
        JZ      INPUT    NOT READY, WAIT
        IN      DATA     GET THE CHARACTER
        RETURN           AND RETURN
```

Example 1.

```
; CONSOLE IN/OUT WITH CONTROLS V78.0          6 MAR 78
CNTIN:   IN    STATUS   ; LOOK FOR INPUT
         ANI   MASK     ; DATA READY?
         JZ    CNTIN    ; NOT YET
         IN    KEYBD    ; READ KEYBOARD DATA
         CPI   14H      ; IS IT CTRL T?
         JZ    CTRLT    ; YES!
         CPI   16H      ; OR CTRL V?
         JZ    CTRLV
         RET            ; NO, RETURN IT
CTRLT:   MVI   A,OFFH   ; SET TTY FLAG
         STA   FLAG
         JMP   CNTIN    ; GET NEXT CHARACTER
CTRLV:   MVI   A,O      ; CLEAR FLAG
         STA   FLAG
         JMP   CNTIN    ; AND CONTINUE
CNTOU:   LDA   FLAG     ; TEST OUTPUT DEVICE
         CPI   O
         JZ    CRTOU
TTYOU:   IN    TTSTAT   ; OUTPUT TO TTY
         ANI   TTMASK   ; WHEN READY
         JZ    TTYOU
         POP   PSW      ; CHARACTER TO
         OUT   TTYPT    ; TTY PORT
         RET            ; AND RETURN
CRTOU:   IN    CRSTAT   ; OUTPUT TO CRT
         ANI   CRMASK   ; WHEN READY
         JZ    CRTOU
         POP   PSW
         OUT   CRTPT
         RET            ; AND ALL DONE!

         END
```

Example 2. The control input and output routines that are hidden away in an "invisible" area of memory. These routines enable the selection of the output device.

Other Applications

Expansions of the same technique can be used to implement enhancements to almost any program. For example, a pair of control characters could be used to implement a pause and continue in any program that continually tests the keyboard during execution. This is handy for slowing things down as they flash by on the TV display.

In another application, after the amount of main memory was doubled in a mini—to make room for more application programs—the extra memory was used as a "mass storage" device for the text editor output and assembler input. The assembler operation was speeded up dramatically. The control input routine was used to direct the text editor output from the usual paper tape "punch" device to the top half of the main memory.

The new control routine took care of placing each output character into the next successive memory location. When the assembler was then loaded, "↑R" was used to "rewind" the source code "tape" by resetting the memory pointer to the start of the buffer. Another control character was used to switch the assembler from the original paper tape reader input to the memory-as-reader routine.

The resulting system would assemble faster than a disk-operating system, at a fraction of the cost of a disk, but of course without the same amount of storage a disk would provide.

Only the I/O driver program locations within the original operating system were known, but then they were all we needed to know to add useful enhancements like these. ■

CT-1024 Terminal Modifications

Expand your cursor control capability, and add a clock.

Fred Cooley
4646 Willis Ave.
Sherman Oaks CA 91403

This article describes two circuit diagrams for the SWTP CT-1024 CRT terminal that I've built and tested on my machine. These modifications will be useful for any CRT terminal device.

Expanded Capability Cursor Control

The heart of the CT-CA computer-controlled cursor board is the 7445 BCD-to-decimal decoder, which breaks down the ASCII control characters to one of ten different outputs. By replacing the 7445 with one or two 74154 demultiplexers, either 16 or 32 outputs may be obtained (see Fig. 1). These control character outputs serve those functions handled by the CT-CA board, and also control other internal functions such as page select, cursor on/off, cursor solid/blinking or any external function.

All outputs can be computer controlled by outputting the proper ASCII control character. Any output to be "held" on is fed to an R/S latch. Connections to the CT-1024 are made using Molex pins over the CT-CA's connector strips J3 and J4. Don't use certain outputs:

CTRL M — CT-1024 internal carriage return
CTRL J — CT-1024 internal line feed
CTRL C — SWTP 8K BASIC ready command
CTRL O — SWTP 8K BASIC back space
CTRL X — SWTP 8K BASIC delete command

CT-1024 TV Clock

An article from *Radio Electronics*, July 1977, "Build This Digital On Screen TV Clock," describes the three IC clock. This modification (see Fig. 2) uses the same basic circuitry; however, the ac reference and horizontal and vertical sync pulses are taken from the CT-1024, and the clock's output is fed along with the terminal's video output to an rf modulator. Thus, no circuitry within the TV itself is touched.

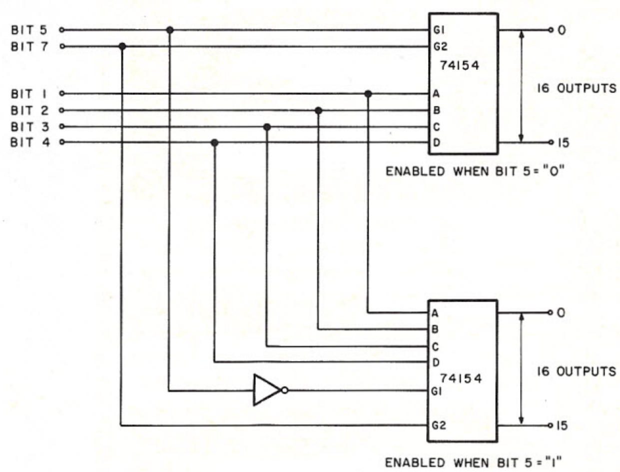


Fig. 1. Cursor control circuit.

Time-setting functions, on/off display and 12/24-hour format are controlled by the expanded cursor control circuit using keyboard control characters to set or reset R/S latch circuits. The MM5318 digital clock IC has seven-segment outputs that can be interfaced to the SWTP 6800 computer as described in *Byte*, November 1977, "Does Anybody Know What Time It Is?"

This circuit requires four connections to the CT-1024. The ac reference is obtained at J-11, pin 5; horizontal sync and vertical sync are taken from IC17, pins 5 and 4, respectively, and the video output is tied to J-10, pin 11. ■

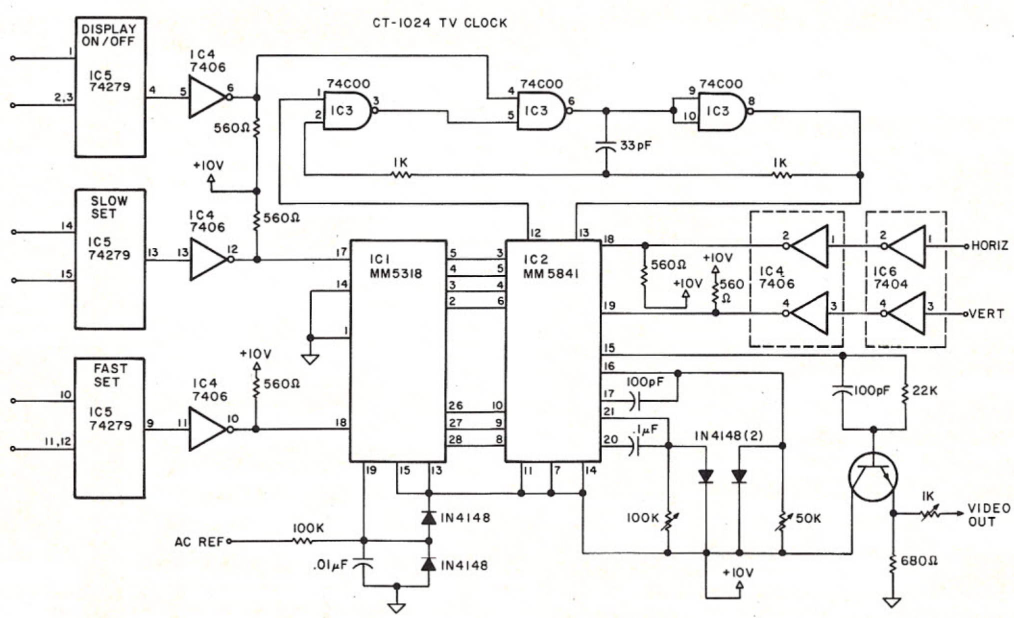


Fig. 2. TV clock circuit.

Edited by Dennis Brisson

MICRO-SCOPE

Pan Am Flies More Efficiently

Pan American World Airways plans to save more than 7,500,000 gallons of fuel a year for its fleet of 747 Jumbo Jets. They expect to realize this savings through the installation of Delco Electronics' new Flight Management System (FMS), a computerized system designed to minimize fuel consumption by regulating throttle adjustments to maintain optimum speed as selected by the pilot in conjunction with pitch control through the plane's automatic pilot system. The FMS takes into account and makes adjustments for the aircraft's constantly changing weight, altitude, outside temperature and other factors. In so doing, the FMS maintains the aircraft at a specified speed and minimized acceleration during maneuvering, thereby reducing unnecessary use of fuel.

The Flight Management Computer, which utilizes Motorola's NMOS LSI circuits, is being installed on 39 Boeing 747s, with six all-cargo 747s targeted for installation of the system at a later date. The airline plans to save approximately $5.55 million per year.

Flying High with Help from TI

Computers are also helping to make the aviation industry more efficient through inventory control. Clients of Inventory Locator Service, Inc., of Memphis, TN, use a Texas Instruments 765 Data Terminal to tie into a central ILS host computer data bank, which inventories over a million available new and used aircraft parts and support items from many suppliers. In this way, the computer system, with built-in acoustic couplers, reduces airplane downtime and helps to get the planes off the ground with speedy, inexpensive inventory searches.

She Talks in Beauty

Who is generally recognized as the first computer programmer?

According to a course brochure from George Washington University, it was the Countess of Lovelace, a lady named Ada who is noted for her work in the middle 1800s with Charles Babbage, the "father of the computer." Her place in computer history has been duly recorded with the introduction of a computer programming language named after her. The new language, called Ada, was recently developed by the U.S. Department of Defense for use in embedded computer applications.

Incidentally, the father of Ada (the person, not the language) was George Gordon Byron, an early nineteenth century English poet, who is also noted for his work with language.

Making the Rounds

The medical data-processing community can now make use of a recently developed information service to aid in the understanding, evaluation and selection of a hospital laboratory computer system. "The MedSy Report on Clinical Laboratory Computer Systems" contains teaching materials, service materials and research support to define the potential and capability of computers in this field and keep users informed of any new developments or changes.

This product results from three years of clinical laboratory systems research from leading American universities and private hospitals. The service includes receipt of a MedSy handbook to help you evaluate and select a system for your particular needs. Subscribers receive updated information supplied on a bimonthly basis. For more information, contact Medical Systems Research, Inc., 2025 N.W. 24th St., Gainesville, FL 32605.

CRT Terminal Users Favor 80 Columns

What is the most popular CRT terminal size?

According to a recent survey conducted by Venture Development Corporation, a Wellesley, MA, consulting firm, users of alphanumeric CRT terminals express a strong preference for 80-column by 24-row displays. The 80-column display has become standardized to the point where only five percent of users surveyed prefer displays with fewer columns, and only nine percent have a preference for more than 80 columns. Although the 132-column display has applications in selected areas, most users are not willing to pay the increased price for a feature which they feel has only marginal value.

However, users expressed a desire for more total characters per display, but felt this should be done through additional rows, not more columns. A 25th row is highly desirable, as are additional rows, for word-processing applications. The 25th row, which has been gaining wider acceptance, is used primarily for monitoring system status and control rather than for display of data.

Test Your ESP

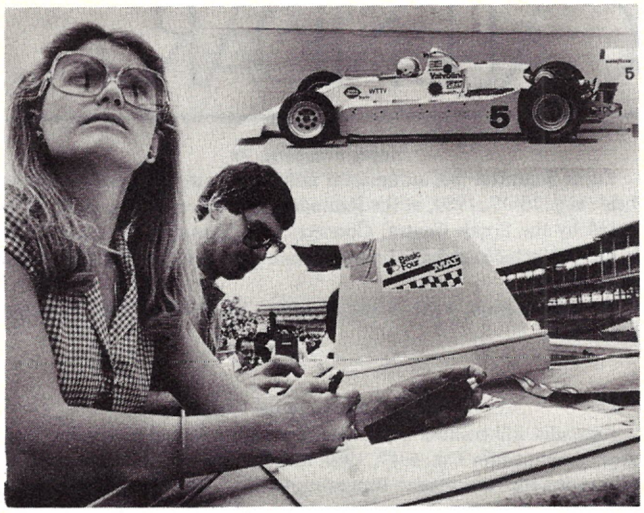
Now you can test your extrasensory powers on your 16K TRS-80 with a new program from Manhattan Software, Inc., PO Box 5200, Grand Central Station, New York, NY 10017. E.S.P. Lab ($9.95) can be used for research into possible extrasensory phenomena, as well as for testing the possibility of telepathy, clairvoyance, precognition and telekinesis. It selects randomly from among a set of five symbols, presenting one symbol at a time on the screen for telepathy experiments. All symbols are programmed in machine language and appear on the screen instantaneously. For clairvoyance and precognition testing, the program selects the symbol before or after the response, prompting only with a question mark on the screen.

A separate section provides a special computer-style test of possible telekinesis. A randomly moving dot is presented on the screen, and the experimenter may attempt to use "mental power" to influence the direction of movement of the dot.

Persons who become proficient at this program are requested not to enter within a 150-mile radius of the Mount St. Helens area.

ENTER: "Nightly News"

Computers may soon take the place of TV sets in bringing the nightly news into the family living room. Beginning this summer, 11 Associated Press members will begin an experiment into the new technology of information retrieval with CompuServe, Inc., a Columbus, OH, computer firm. Home personal computer owners will be able to obtain newspaper information by dialing special telephone numbers. The cost will be $5 an hour to access news, sports, business and feature data provided by the newspapers and the AP. CompuServe also provides computer programming and games through an existing personal computing network available in more than 250 cities. Each newspaper will participate for a six-month period, providing news, features and advertising material in its community.



Laps driven by Al Unser were entered manually and stored in the Basic Four System 410 small-business computer, which provided racing strategy for the Unser racing team at the 1980 Indianapolis 500 Auto Race.

Computer Pit Stop

Al Unser's pit crew at this year's Indianapolis 500 Auto Race included the first computer ever to be used at Indy. The Basic Four System 410 small-business computer provided Unser with a computerized management information system. Designed for use in strategy planning for the race, the computer endured 100 degree temperatures, typical Indiana humidity and crackling spikes of electromagnetic interference to monitor the performances and positions of the 15 leading contenders.

For each car, the 410 computed the position in the race, total laps completed, lap number of last pit stop, track condition (yellow or green), yellow laps since last pit stop, total laps since last pit stop and total number of pit stops. Additionally, it tracked the fuel consumption of the car driven by Unser.

This data enabled Unser's racing team management to make split-second decisions to schedule pit stops under the most advantageous conditions. Electronic sensing devices and high-speed calculators are routinely used on the racing circuit, but the introduction of the computer represented an all-new dimension in race-management sophistication.

The computer consisted of a CPU with 96K bytes of memory, 14 megabytes of magnetic disk information storage capacity, three video display terminals, a 160 cps printer and a 9.2 megabyte tape cartridge for system backup and additional data storage capacity. The system functioned throughout the entire 500-mile race, keeping flawless track of the 15 selected cars and accurately recording Johnny Rutherford's first place finish in the race.

Unfortunately, the car driven by Unser, the only winner of auto racing's triple crown and three-time Indy 500 champ, did not match the computer's performance. Unser finished 27th in a field of 33, having to leave the race after 33 laps with a blown engine.

COMING NEXT MONTH

Next month's issue will place particular emphasis on the Commodore PET.

Among the articles and topics covered will be:

- A review of the new Video Interface Computer (VIC) from Commodore.
- Adding 16K to the PET.
- Interfacing PET and the H14 printer.
- Animated graphics.
- The conclusions to the "I/O Expander" and "IEEE 488" articles.

Even if you're not a PET user, there'll be something for you.

CALENDAR

Super Scarafest '80

Super Scarafest '80, an amateur radio and computer festival, will be held Aug. 16–17, 1980, at the Ramada Inn in North Haven, CT. Sponsored by the South Central Connecticut Amateur Radio Assoc., this event will include exhibitor booths, a ham and computer flea market and an auction.

For information, write: Super Scarafest '80, PO Box 5265, Hamden, CT 06518.

Eighth World Computer Congress

Australia will co-host with Japan the Eighth World Computer Congress scheduled for Oct. 6–17, 1980, and conducted by the International Federation for Information Processing (IFIP). The congress commences in Tokyo, Japan, Oct. 6–9 and concludes in Melbourne, Australia, Oct. 14–17. An exhibition of hardware and related services together with submission of papers and discussions will be scheduled in each location. A single registration fee will cover attendance at both locations.

For information, contact: The Eighth World Computer Congress, IFIP Congress '80, GPO Box 880G, Melbourne, Victoria, Australia 3001.

Four Tutorials Precede Compcon Fall '80

Four pre-conference tutorials will be presented on Monday, Sept. 22, 1980, immediately preceding Compcon Fall '80, sponsored by the IEEE Computer Society. "Distributed Processing," the theme of the conference, will be the unifying thread of the tutorials. Topics to be presented are: "Local Computer Networks," "An Overview of Distributed Processing," "Communication Technology in the 80's" and "Distributed System Design." Compcon Fall '80 is Sept. 22–25, 1980 at the Capital Hilton Hotel, Washington, DC.

For more information, write: Compcon Fall '80, PO Box 639, Silver Spring, MD 20901, or call 301-589-3386.

Conferences for Computer Use in Small Business

Three regional computer conferences will be held on the theme: Thinking Small—Using Small Computers to Increase Business Productivity. They will be held Sept. 18–21, 1980, in Washington, D.C.; Oct. 16–19, 1980, in Chicago, IL; and Nov. 20–23, 1980, in Boston, MA.

Sponsored by The Information Exchange, each conference program will be a four-day program designed to explore the opportunities presented by small computers for improved productivity in small businesses.

There will be a concurrent exposition at each location. For further details, contact Kendall Burroughs, at The Information Exchange, 1730 North Lynn Street, Suite 400, Arlington, VA 22209, 703-521-6209.

Mid-Atlantic Computer Show

The Mid-Atlantic Computer Show will be held at the D.C. Armory/Starplex, Washington, D.C., on Sept. 18–21, 1980. General adult admission is $5 to this exposition featuring small and medium-sized business systems, scientific, engineering computers and microcomputers. For information, contact: National Computer Shows, 824 Boylston St., Chestnut Hill, MA 02167, 617-739-2000.

Conference for Consumer Electronics Instructors

The first special Conference for Consumer Electronics Instructors will be held as part of the NESDA/ISCET Convention, Aug. 18–23, at the Galt House, Louisville, KY. The concept of the Instructors Conference is to upgrade educational techniques for electronics instructors and to encourage the development of each curriculum to meet today's rapidly changing technology.

Important events at the week-long NESDA/ISCET Convention include the "Electronics Derby" trade show, the National Service Conference to discuss industry problems, business management and technical sessions and a special conference for electronics instructors. Various electronics firms will be sponsoring local trips, dinners, lunches and cocktail parties. For more information or registration blanks, write or call NESDA, 2708 West Berry St., Ft. Worth, TX 76109, 817-921-9061.

New Jersey Computer Show

The 1980 New Jersey Personal Computer Show and Fleamarket (NJPCS) will be the first home and hobby computer show ever held in Northern New Jersey. The show is Sept. 27 and 28 at the Holiday Inn (North) at Newark International Airport (NJ Turnpike Exit 14). Featured will be in an indoor commercial exhibit area, a large outdoor fleamarket and user group meetings/forums on the TRS-80, PET, Apple, Heath and other popular systems. For additional information, write: NJPCS, Kengore Corp., 9 James Ave., Kendall Park, NJ 08824.

AED Workshops

The Association for Educational Data Systems is offering a series of workshops especially designed for administrators, educators and computer professionals interested in computers in education.

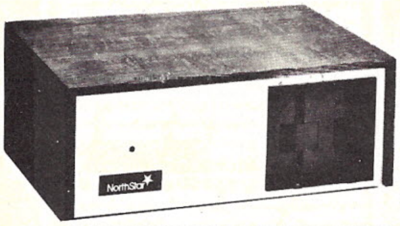
Workshops offered include; Computers in Elementary Education, Aug. 21–22, 1980, in Denver, CO; Programming with Pascal—Learning When and Why, Sept. 25–26 in St. Louis, MO; Computerized Data Base Management, Oct. 9–10 in St. Louis, MO; Computers as Effective Tools for Education—The Evidence, Oct. 23–24 in Des Moines, IA; Word Processing, Nov. 7 in Wichita, KS; Design and Development of Computer-Based Instructional Materials, Nov. 12–13 in Orlando, FL; and Micro-Mini Computers, Personal Computers, and the Development and Evaluation of Educational Programs in Computer Science and Data Processing, Feb. 12–13 in Orlando, FL and Mar. 12–13 in St. Louis, MO.

For information, call: 202-833-4100.

If North Star or Cromemco offer it . . .

WE HAVE IT!!

Immediate Delivery at Discount Prices



NORTH STAR Horizon® 2

32K Double Density
Assembled and Tested
List $3095

ONLY $2619

North Star KIT products have been discontinued. MiniMicroMart HAS INVENTORY of most items!

KITS		ASSEMBLED	
HORIZON 1 16K, DD	**$1474**	**HORIZON 1**, DD	**$2279**
32K, DD, List $1999	**1684**	32K, QD, List $2995	**2539**
32K, QD, List $2199	**1869**	**HORIZON 2**, 32K, DD	**$2619**
HORIZON 2, 16K, DD	**$1824**	32K, QD, List $3595	**3049**
32K, DD, List $2399	**2034**	48K, DD, List $3590	**3039**
32K, QD, List $2779	**2359**	48K, QD, List $4090	**3469**
		64K, DD, List $3830	**3239**
		64K, QD, List $4330	**3669**

NORTH STAR APPLICATIONS SOFTWARE

(Exclusive for use with North Star Disk Systems — specify Double or Quad Density)

NORTHWORD, List $399	**$339**
MAILMANAGER, List $299	**249**
INFOMANAGER, List $499	**419**
GENERALLEDGER, List $999	**799**
ACCOUNTSRECEIVABLE, List $599	**499**
ACCOUNTSPAYABLE, List $599	**499**

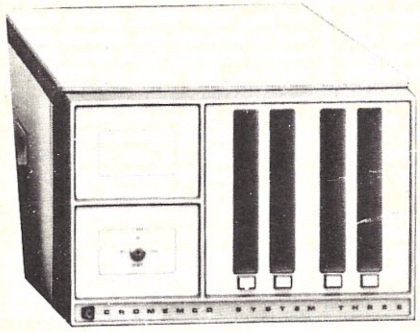
NORTH STAR HARD DISK HD-18

18 megabytes, plugs into parallel port of North Star Horizon. Utilizes tried-and-proven 14" Century Data Marksman. List $4999. **OUR PRICE $4199**

NORTH STAR MDS-A — Double (or Quad) Density Disk System, Kit, List $799 . **OUR PRICE $669**
Assembled and Tested, List $899 **SPECIAL $719**

NORTH STAR MEMORY BOARDS

16K Dynamic RAM (RAM-16-A/A), Assembled, List $499 ..... **$420**
Kit, List $449 ..... **SPECIAL $299**
32K (RAM-32/A), Assembled, List $739 ..... **$620**
Kit, List $669 ..... **ONLY $499**



CROMEMCO System 3

with 64K of RAM
List $6990

OUR PRICE $5890

CROMEMCO SYSTEM 2 — Now features dual-sided drives — double the capacity. Similar to System 3, except features dual, double-sided mini floppy disk drives. List $3990 ..... **ONLY $3390**

Z-2 COMPUTER SYSTEM (can be rack mounted), List $995 ..... **$845**

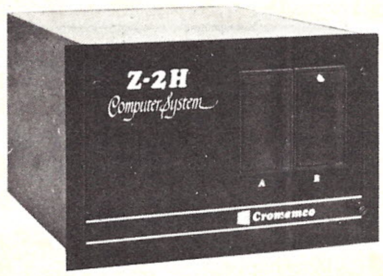
SINGLE CARD COMPUTER — SCC-W
4 MHz. List $450 ..... **$382**

***NEW* COLOR GRAPHICS INTERFACE — SDI**
List $595 ..... ***OUR PRICE ONLY $505***

CROMEMCO HDD – 11/22-megabyte Hard Disk for use with existing systems. DMA controller. Transfer rate of 5.6 megabytes/second.

HDD-11, List $6995 ..... **OUR PRICE ONLY $5939**
HDD-22, List $11,995 ..... **$10,189**

CROMEMCO Z-2H Full 11-megabyte Hard Disk system. Fast Z-80A 4 MHz processor, two floppy disk drives, 64K RAM memory, RS232 special interface, printer interface, and extensive software available. List $9995



OUR PRICE $8489

SHIPPING AND INSURANCE: Add $15 or Horizons, $2.50 for Boards and Software. Hard Disk Systems and Cromemco systems shipped freight collect. Advertised prices are for prepaid orders. Credit card and C.O.D. 2% higher. Deposit may be required on C.O.D. All prices subject to change and offers subject to withdrawal without notice.

— WRITE FOR FREE CATALOG —

✓275

MiniMicroMart, Inc.

1618 James Street, Syracuse, NY 13203 (315) 422-4467 TWX 710-541-0431

master charge THE INTERBANK CARD

VISA

CLASSIFIEDS

Classified advertisements are intended for use by persons desiring to buy, sell or trade used computer equipment. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows up to 5 lines of about 35 characters per line, including spaces and punctuation. The $10 box allows up to 10 lines. Minimize use of capital letters to save space. No special layouts allowed. Payment is required in advance with ad copy. We cannot bill or accept credit.

Advertising text and payment must reach us 60 days in advance of publication (i.e., copy for March issue, mailed in February, must be here by Jan. 1). The publisher reserves the right to refuse questionable or inapplicable advertisements. Mail copy with payment to: **Classifieds, Kilobaud Microcomputing, Peterborough NH 03458.** Do not include any other material with your ad as it may be delayed.

For Sale: D. C. Hayes Micromodem for Apple II. Almost new, works fine! Only $320. Contact: H. Rothman, 218 Huntington Road, Bridgeport, CT 06608. 203-579-0472.

Super Elf computer with supply, cabinet, memory saver, CMOS RAM, low address display, expandable ($185 value), $90. Peter Marcus, 10225 Coral Way, Apt. B122, Miami, FL 33165.

Sale: Xitan Z80 system 48K RAM, 2 Micropolis disks, 630K online, Lear Siegler ADM-3A terminal, software, manuals, like new, $4100. Call 914-357-8482.

For Sale: SWTP 6800 complete system. Including: 24K memory, EPROM programmer MP-S, SWTBUG, CFM-3 (ROM), SWTP 4K and 8K BASICs, TSC BASIC, Assembler/Editor, Disassembler, CT-64, AC-30, MPC, PR-40, manual, spare chassis and cards, much more. A $2000 value for only $1000. Contact: Les Asher, 203-847-9000.

Sale: Look into your SS-50 micro with Percom's Electric Window, includes 2516 EPROM. $210 ppd., USA. Victor Kola, 2360 LK George Dr., Anchorage, AK 99504.

PDP-11/20 minicomputer—20K core memory total with expansion box (Plessey) and DLV-11 console interface. Full set of documentation, diagnostic paper tapes, Pal-11, ED-11 and BASIC. $1700. Bill Beckett, 77 Clinton St., Framingham, MA 01701 or 617-879-0268.

For Sale: Heath H8 8K memory boards. Four at $110 each ppd. Removed due to memory expansion. Louis Berry, 2259 N. Marter Ct., Simi Valley, CA 93065. (805) 522-3599, eves.

OSI C1P-MF-20K RAM. 5¼" disk drive. Much software. 1 yr. old. Over $1300 invested. Moving to larger system. Asking $950 or ? D. Lockwood, 9045 Park Meadows Dr., Elk Grove, CA 95624.

For Sale: 16K Level II TRS-80 w/keypad, new, under warranty, $699. Contact John Abraham or Joe Gutierrez, PO Box 3945, McAllen, TX 78501. 512-687-2329 or 512-682-6731.

Ohio Scientific C1P, extra documentation and demo tape, $295. Ten high quality C1P programs, Backgammon (8K), Super Monitor, Starfighter, Slashball, Breakthrough, Hangman, Concentration, Tank Battle, Killerbot and Compress; $45. Goldstar TV/monitor; $85. All like new, all for $395. I pay postage (USA). Write: Jerry Travis, 8533 Pacific Hwy SE, Olympia, WA 98503.

For Sale: Data Processing Service Bureau in Colorado mountain and ski town. Sale includes Pertec PCC 2000 computer, NEC printer, desk, established customers, prospect list. Software for Word Processing, G. Ledger, Payroll, Job Cost, Acc. Receivable, Acc. Payable, Mailing Lists. Contact Ken Hampton, PO Box 282, Crested Butte, CO 81224 or call 303-349-6203.

Altair dual disk drives with controller and EPROM boot boards. Drives just back from factory overhaul. All for $1500. Original cost over $3000. Dick Whipple, 305 Clemson Dr., Tyler, TX 75703. 214-561-1648.

SWTPC/S100 PR-40, $180; 3 Godbout Econoram II, $110 ea.; adapter SS50 bus to S100 bus, see Nov. '78 *Kilobaud*, $15. Call Mike Witte, work 901-345-6222 or home 901-795-5122.

For Sale: Micromation Megabox. Two double-density drives. Disk controller for S100 bus. New/never used. Cost $2300. Will sell for $1800 or best offer. Jim Fritz, 1413 Harmony Lane, Annapolis, MD 21401, 301-757-7019.

For Sale: AJ 841 I/O serial printer includes everything needed to use with TRS-80. Also lowercase kit. R. Clore, 2711 Hatch Rd., Jackson, MI 49201. 517-563-2978.

For Sale: S-100 equipment. BIOTECH CGS-80 color graphics board, 7K—2114 & 8085 on board—$275. 16K—250 ns. 2114 RAM board—$275. 8K Piceon EPROM programmer board w/8-clean 2708s—$200. North Star MDS disk system w/10 disks; DOS, BASIC, XEK assembler and much more—$500. Five 8K—2102 RAM boards—$80 ea. 16-slot S-100 mainframe w/15 connectors, Mits front panel, 2SIO serial interface and CPU—$350. Bonus—$1950 takes everything plus a Datapoint 3300P, 300 baud, thermal printer; 80 column with TI silent 700 paper. Guarantee all for 30 days. Dave Valliere, 207 Edgewood Dr., Wilmington, DE 19809. 302-764-7210.

Leedex 100 Video Monitor, excellent condition, $100 or best offer. C1P tapes, assorted OSI and Aardvark, retail $128—all for $90. Barry Beal, RFD Box 160, Machias, ME 04654.

For Sale: Unused Polymorphic System 8813 with 32K RAM memory with floppy-disk drives, printer interface and Abern-Sopher Multiwriter III. System has Canadian import tax paid. Offers for complete system to: Bishop Management, #8—825 McBride Blvd., New Westminster, B.C., Canada, V3L 5B5. 604-525-8148.

Wanted: IC chip, type #SW-10667, used in Diablo 1622 daisy-wheel printer. Jim Jamison, 2304 Tucker, Suncoast, N. Ft. Myers, FL 33903.

Soroc IQ 140 terminal in original box, $995. IBM Selectric 731 I/O, $450. Bob Flaming, 6519 Jetta Ave., Bakersfield, CA 93308. 805-399-8968.

Amateur radio—Macrotronics TRS-80 RTTY system—M80, M800 hardware. Software—tapes—manuals & bulletins, $130. Write: Al Hubbard, 5705 Junonia, SPV, Fort Myers, FL 33908.

For Sale: Datapoint 3300P thermal printer, 80 column, RS-232 serial input at 300 baud. Uses TI 745 paper. Case is 15" × 14" × 5" and color matches "Apple." Quiet, reliable and attractive printer. 30 day guarantee. $500, freight collect, $550 w/Apple serial interface. Dave Valliere, 207 Edgewood Dr., Wilmington, DE 19809. 302-764-7210.

Not to worry, sir. I've seen worse than this.

CORRECTIONS

The left brackets ([) that appear in the "Orbiter" listings (May 1980, pp. 112, 113) should be up-arrows.

For 16K machines, the last two digits in the "POKE" line of Ronald Foulis's letter (June 1980, p. 225) should be the three digits 127.

The digits in line 120 of Al Joffe's program listing on page 32 of the September 1979 issue should be 9144, not 9114.

Gary Sabot, author of "You Name It!" (May 1980), writes: In the two years since I wrote the article, the version of BASIC I used has become obsolete. In order for the program to run on any version of Microsoft BASIC (such as MBASIC and Radio Shack Level II), a small change must be made. Wherever the old program has "MID(. . ." you must substitute "MID$(. . .". Lines 190, 200 and 210 must be changed in this way.

Allan Domuret writes that there is a typo in Example 2 of his article "Uppercase Lowercase Utility for the TRS-80" (March 1980). The POKEs to decimal memory references 16512, 16513 and 16514 should read 16408, 16409 and 16410, respectively. The hex references to &H4018, &H4019 and &H401A are correct as shown.

Author of "Lowercase for the TRS-80," Steven Wexler, let us know that there is an error in his listing. The next to the last line should read, in part: 7FFA D2,7D,04. The second element is D2, not 02.

This is not a correction; it is just a note that the list of sources of educational software on page 105 of "Bringing Microcomputers into Schools" (June 1980) was not meant to include all sources.

Chesney Twombly, author of "8080 Program Loader/Relocator" (April 1980), wants to add the following note to his article: The TSC 8080 Relocator program will run on an H8 using CONOPS if you remember to change the addresses of external routine jumps to read as follows:

```
431F C3 DB 6E  MONITR  JMP   MONIT
4322 C3 1D 6C  INCH    JMP   INCHR
4325 C3 67 20  OUTCH   JMP   WCHAR
```

Robert Penoyer, author of "Accurate Voltage Dividers" (April 1980), has informed us that the last sentence of the second complete paragraph on page 143 should finish: "3900 Ohms and 12 Ohms," not "3900 Ohms and 1200 Ohms."

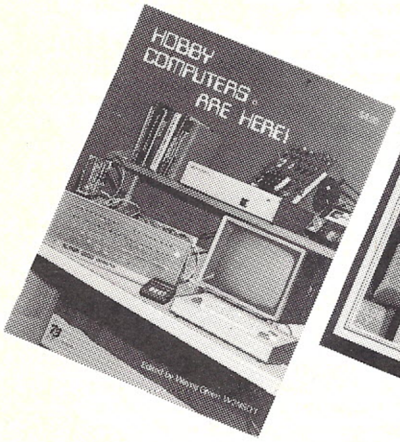
kb microcomputing book nook

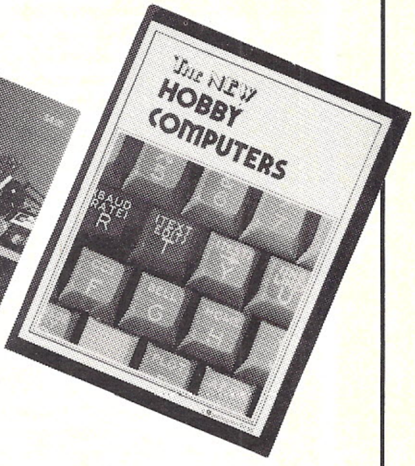
2 NEW BOOKS FROM KB MICROCOMPUTING

• **40 COMPUTER GAMES**—BK7381—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. $7.95*

• **UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS** –BK7382–A valuable addition to your computing library. This two part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of the new microcomputing hobby. Well known authors and well structured text helps the reader get involved in America's fastest growing hobby. $10.95*

INTRODUCTORY





• **HOBBY COMPUTERS ARE HERE!**–BK7322–If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, and the basics of programming. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

• **THE NEW HOBBY COMPUTERS**–BK7340–This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards . . . and much, much more! Don't miss this tremendous value! Only $4.95.*

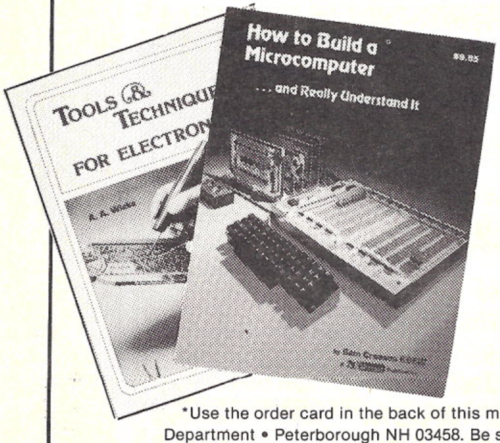
INTRODUCTION TO MICROCOMPUTERS (VOL. 0→III)

• **AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0**–BK1130–The Beginner's Book–Written for readers who know nothing about computers–for those who have an interest in how to use computers–and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. $7.95.*

• **VOL. I**–BK1030–**2nd Edition completely revised.** Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. $12.50.*

• **VOL. II**—BK1040 (with binder)—$30.00* —Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips.

• **VOL. III**—BK1133 (with binder)—$20.00.* Contains descriptions of all support devices that can be used with any microprocessor.



• **HOW TO BUILD A MICROCOMPUTER–AND REALLY UNDERSTAND IT**–BK7325–by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.*

• **TOOLS & TECHNIQUES FOR ELECTRONICS**–BK7348–is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

SPECIAL INTERESTS

● **TRS-80 DISK AND OTHER MYSTERIES**—BK1181—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

● **INTRODUCTION TO TRS-80 GRAPHICS**—BK1180—by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use. $8.95.*

● **MICROPROCESSOR INTERFACING TECHNIQUES**—BK1037—by Austin Lesea & Rodnay Zaks will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100. $15.95.*

● **MICROPROCESSOR LEXICON—ACRONYMS AND DEFINITIONS**—BK1137—Compiled by the staff of SYBEX is a convenient reference in pocket size format. Sections include acronyms and definitions, part numbers and their definitions, S-100 signals, RS232 signals, IEEE 499 signals, microcomputer and microprocessors, JETDS summary (military) and a code conversion table. $2.95.*

● **MICROPROCESSORS FROM CHIPS TO SYSTEMS**—BK1036—by Rodnay Zaks is a complete and detailed introduction to microprocessors and microcomputer systems. No preliminary knowledge of computers or microprocessors is required to read this book, although a basic engineering knowledge is naturally an advantage. Intended for all wishing to understand the concepts, techniques and components of microprocessors in a short time. $9.95.*

MONEYMAKING

● **HOW TO MAKE MONEY WITH COMPUTERS**—BK1003—In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

● **HOW TO SELL ANYTHING TO ANYBODY**—BK7306—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. $2.25.*

● **FREELANCE SOFTWARE PUBLISHING**—BK1179—by B. J. Korites. "This book is about money and how to make it by writing and selling computer programs," (author's foreword). If you have the skills to write a saleable program, you now need to acquire the skills to sell that program. This compact book comprehensively covers the entire publishing process and many aspects of software salesmanship. $14.95.*

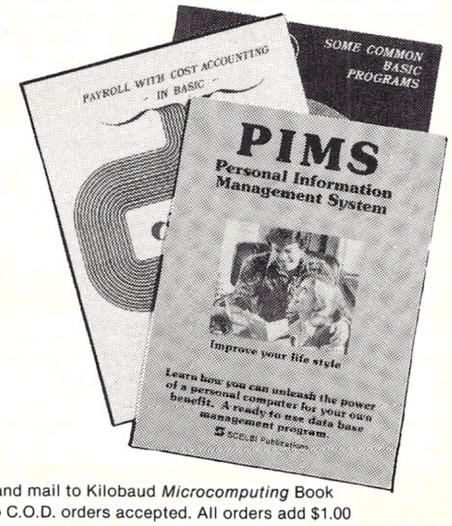
● **THE INCREDIBLE SECRET MONEY MACHINE**—BK1178—by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. $5.95.*

BUSINESS

● **PAYROLL WITH COST ACCOUNTING—IN BASIC**—BK1001—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in **CBASIC**). $20.00.*

● **SOME COMMON BASIC PROGRAMS**—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $12.50 paperback.*

● **PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM**—BK1009—Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $11.95.*



*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

—PROGRAMMING & COOK BOOKS—

—Z80—

● **PROGRAMMING THE Z-80**—BK1122—by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **Z-80 ASSEMBLY LANGUAGE PROGRAMMING**—BK1177—by Lance A. Leventhal. This book thoroughly covers the Z80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $12.50.*

● **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1045—by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $15.95.*

—6502—

● **PROGRAMMING THE 6502 (Second Edition)**—BK1005—Rodnay Zaks has designed a self-contained text to learn programming, using the 6502. It can be used by a person who has never programmed before, and should be of value to anyone using the 6502. The many exercises will allow you to test yourself and practice the concepts presented. $12.95.*

● **6502 APPLICATIONS BOOK**—BK1006—Rodnay Zaks presents practical-application techniques for the 6502 microprocessor, assuming an elementary knowledge of microprocessor programming. You will build and design your own domestic-use systems and peripherals. Self-test exercises included. $12.95.*

● **6502 ASSEMBLY LANGUAGE PROGRAMMING**—BK1176—by Lance A. Leventhal. This book provides comprehensive coverage of the 6502 microprocessor assembly language. Leventhal covers over 80 programming examples from simple memory load loops to complete design projects. Features include 6502 assembler conventions, input/output devices and interfacing methods, and programming the 6502 interrupt system. $12.50.*

● **6502 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1055—by Robert Findley. This book introduces the BASIC language programmer into the realm of machine-language programming. The description of the 6502 structure and instruction set, various routines, subroutines and programs are the ingredients in this cookbook. "Recipes" are included to help you put together exactly the programs to suit your taste. $12.95.*

—8080 / 8080A—

● **8080A/8085 Assembly Language Programming**—by Lance Leventhal—BK1004—Assembly language programming for the 8080A/8085 is explained with a description of the functions of assemblers and assembly instructions, and a discussion of basic software development concepts. Many fully debugged, practical programs are included as is a special section on structured programming. $12.50.*

● **8080 PROGRAMMING FOR LOGIC DESIGN**—BK1078—Ideal reference for an indepth understanding of the 8080 processor. Application-oriented and the 8080 is discussed in light of replacing conventional, hard-wired logic. Practical design considerations are provided for the implementation of an 8080-base control system. $9.50.*

● **8080 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1102—If yu have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 8080 user. $12.95.*

—6800—

● **6800 PROGRAMMING FOR LOGIC DESIGN**—BK1077—Oriented toward the industrial user, this book describes the process by which conventional logic can be replaced by a 6800 microprocessor. Provides practical information that allows an experimenter to design a complete micro control system from the "ground up." $9.50.*

● **6800 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1075—Like its culinary cousin, *The 8080 Gourmet Guide*, this book by Scelbi Computing and Robert Findley describes sorting, searching and other routines—this time for the 6800 user. $12.95.*

—COOK BOOKS—

● **CMOS COOKBOOK**—BK1011—by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $10.50.*

● **TVT COOKBOOK**—BK1064—by Don Lancaster. Describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.*

● **TTL COOKBOOK**—BK1063—by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter and a digital tachometer. $9.50.*

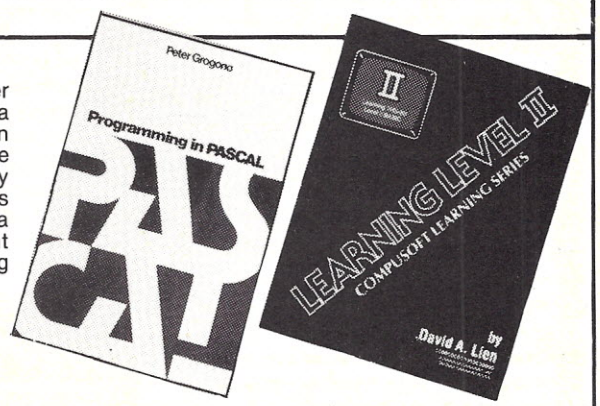
● ***MICROCOMPUTING* CODING SHEETS** ***Microcomputing***'s dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4–6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

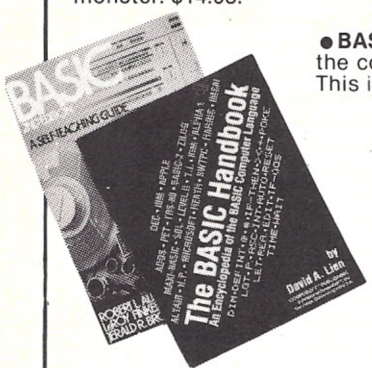
BASIC AND PASCAL



● **PROGRAMMING IN PASCAL**—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language; it is therefore suitable for an introductory course. $12.95.*

● **THE BASIC HANDBOOK**—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

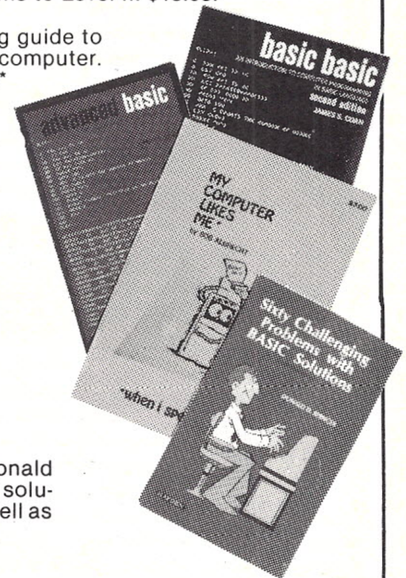
● **LEARNING LEVEL II**—BK1175—by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*



● **BASIC NEW 2ND EDITION**—BK1081—by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $6.95.*

● **BASIC BASIC (2ND EDITION)**—BK1026—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $9.45.*

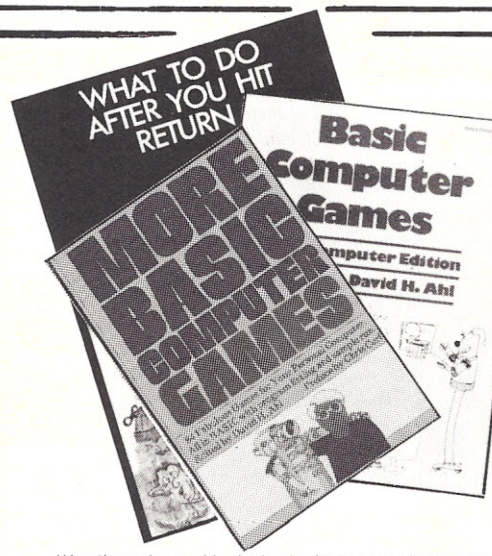
● **ADVANCED BASIC**—BK1000—Applications, including strings and files, coordinate geometry, area, sequences and series, simulation, graphing and games. $9.65*.



● **MY COMPUTER LIKES ME . . . WHEN I SPEAK BASIC**—BK1039—An introduction to BASIC . . . simple enough for kids. If you want to teach BASIC to anyone quickly, this is the way to go. $2.00.*

● **SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS (2nd Edition)**—BK1073—by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

GAMES



● **WHAT TO DO AFTER YOU HIT RETURN**—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

● **BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

new!

● **MORE BASIC COMPUTER GAMES**—BK1182—edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games*. Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. $7.50.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

LETTERS

(from page 23)

of Parsing, Translation, and Compiling, Volume II: Compiling, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1973 (in particular chapter 11 on Code Optimization).

● Baer, J. L., *Graph Models in Programming Systems,* in "Current Trends in Programming Methodology, Volume III, Software Modeling" (K. M. Chandy and R. T. Yeh, editors), Prentice-Hall, Inc. Englewood Cliffs, NJ, 1978.

● Schaefer, M., *A Mathematical Theory of Global Program Optimization,* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1973.

In response to Mr. Spearman: Listings 5-7 illustrate six programs run under WATFIV-S, G Level FORTRAN IV and PL/I on an IBM 4331. (I would like to thank Larry Oliver of the UMR Computer Center for his assistance.) Similar time savings should be realized when this technique is applied to most compilers (be they on a micro, a mini, or a maxi) *if* it is able to recognize a constant subscript. If the compiler is not able to determine that the subscript is a constant, then it will treat this array reference the same as it would treat a reference subscripted with a variable.

Warren A. Harrison
Computer Science Dept.
University of Missouri
Rolla, MO 65401

Correction

Before submitting my article, "A Micro for the Eighties" (May 1980), I talked by telephone with a dealer confirming the then-current prices for AlphaMicro products. Since these have been in effect for 18 months, I was concerned about a price increase. At no time did this dealer or another to whom I talked advise me of any change in policy concerning the sales of equipment by AlphaMicro.

I have since called two dealers and AlphaMicro to find that AlphaMicro made a change in marketing policy effective January 1, 1980. Primarily to enhance total system reliability, AlphaMicro Systems no longer sells the AM-100 CPU boards or the disk control boards as separate items. One can only purchase a complete system from AlphaMicro. The minimal system is an AM-100 with double density floppy (but without terminals or printers). The base price is approximately $10,800.

Some dealers still have AM-100 CPU boards, ordered before the policy change, which they will sell separately for those persons wanting to upgrade an 8080 or Z-8 system.

I am informed that the policy created no small measure of dealer resistance and that many dealers have expressed opposition to the policy. I do not know if this will result in a change—I rather doubt that it will.

Some people at AlphaMicro tell me that the new policy is primarily the result of the introduction of a new and faster CPU called the AM-100-T. Apparently, this CPU stretches most S-100 motherboards beyond their capabilities. The introduction of the new policy, while making it much more expensive to upgrade to an AlphaMicro system, results in AlphaMicro supplying a complete system which they know will work, instead of struggling with users who have placed an AlphaMicro CPU on a substandard motherboard and with inferior memory.

I deeply regret the inaccuracies appearing in the article. I thought I was going out of my way to avoid problems such at this. Apparently, I did not go far enough or ask the right questions. Frankly, it never dawned on me that AlphaMicro would stop selling individual CPU boards. Again, my apologies.

Wm. C. Welborn, Jr.
Evansville, IN

Praise for Pascal

I was pleased to see the article, "An Introduction to Pascal," by Jim Gagne in the June 1980 issue. Besides introducing the language, it also contains a good sales pitch on the benefits of a structured language.

I feel strongly that new users of microcomputers should be learning Pascal as their first computer language. This would avoid all the bad programming habits that people pick up when starting out with an old-style language, such as BASIC. What is really needed here is a simple low-cost computer with Pascal in ROM. This would mean that whenever you switched the machine on, there would be Pascal, ready to run.

Unfortunately, most of the versions of Pascal that are available for microcomputers are compilers, which require floppy disks and extensive memory. For the beginning programmer, an interpreter is much better because it executes the program lines directly, and requires much less memory. It will be slower, of course, but that is not normally a problem.

As an experiment, I recently wrote an interpreter for Pascal, just to see if it could be done. I simplified standard Pascal slightly, to make it easier to interpret. In particular, there must be one statement per line, and semicolons are not used to separate statements. The indentation of lower-level statements is compulsory.

I think that having an interpreter such as this on a small appliance-type microcomputer would be a great advance in hobby computing. The version I have now is written in assembly language for a time-sharing computer, but I intend to do it again for my Motorola 6802 microcomputer.

J. Gary Mills
Winnipeg, Man.
Canada

Love That TRS-80

The April 1980 "Publisher's Remarks" indicated interest in the comments of users of microcomputers for business operations.

Our company uses a TRS-80 (Model 1) with lowercase mod, 48K RAM, standard 35-track disk drives and a daisy wheel printer.

Although our business volume is suitable for a minicomputer system, we prefer our TRS-80 for a number of reasons.

Most important, we can write our own programs, giving us the flexibility to produce and format data as we wish, and to alter the output as our needs change.

The other reasons have mainly to do with cost savings and convenience. For instance, our office payroll and related expenses have fallen $24,000/annum, and service is available through a maintenance contract with a local firm. Also, programs such as Electric Pencil enable us to use it for more functions than we had contemplated.

We did experience some reliability problems. Substituting NEWDOS for TRSDOS solved most of them. Others appear to be caused by such things as poor connections developing at the edge card connectors. After some experience we have been able to recognize the causes and eliminate most problems. We notice an increasing number of reliability aids, such as data separators, on the market and we will no doubt be continually upgrading our equipment.

W. K. Wells
Scarborough, Ontario
Canada

REMARKS

(from page 7)

Opportunity Knocketh

Software and hardware may come and go, but one thing goes on, regardless of the changes, and that's publishing. The faster the changes happen, the more need there is for more magazines and books to support the changes. This explains the growth we've experienced with *Kilobaud Microcomputing* and *80 Microcomputing* magazines.

Since it takes people to make this growth happen and continue, we have career spots open, and no end in sight to the projected growth. If you have any background or desire to apprentice in editing, technical editing, advertising sales, circulation, product testing, programming or any of the other facets of programming, you should let me know. Send me a letter telling me what you think you can do for us and what aspects of your background and experience substantiate that expectation.

Ad Policy

We've received several reader complaints about the recent Interlude ads. This was not unexpected. But we figured that since most readers of this magazine were presumably knowledgeable about sex, censorship of the ad would achieve little.

P.O. Box 4430S, Santa Clara, CA 95054
Will calls: 2322 Walsh Ave.
(408) 988-1640

Quest ELECTRONICS

✓44

NEW PRODUCTS!

Super Color S-100 Video Kit $129.95

Expandable to 256 x 192 high resolution color graphics. 6847 with all display modes computer controlled. Memory mapped. 1K RAM expandable to 6K. S-100 bus 1802, 8080, 8085, Z80 etc. Delivery January '80.

1802 16K Dynamic RAM Kit $149.00

Expandable to 32K. Hidden refresh w/clocks up to 4 MHz w/no wait states Addl. 16K RAM **$63**

Elf II Adapter Kit $24.95

Plugs into Elf II providing Super Elf 44 and 50 pin plus S-100 bus expansion. (With Super Expansion). High and low address displays, state and mode LED's optional **$18.00**.

Gremlin Color Video Kit $69.95

32 x 16 alpha/numerics and graphics; up to 8 colors with 6847 chip; 1K RAM at E000. Plugs into Super Elf 44 pin bus. No high res. graphics.

Quest Super Basic

Quest, the leader in inexpensive 1802 systems announces another first. Quest is the first company worldwide to ship a **full size Basic** for 1802 systems. A complete function **Super Basic** by **Ron Cenker** including floating point capability with scientific notation (number range $\pm.17E^{38}$), 32 bit integer ±2 billion; Multi dim arrays; String arrays; String manipulation; Cassette I/O, Save and load, Basic, Data and machine language programs; and over 75 Statements, Functions and Operators.

Easily adaptable on most 1802 systems. Requires 12K RAM minimum for Basic and user programs. Cassette version in stock now. ROM versions coming soon with exchange privilege allowing some credit for cassette version. New improved version with improved speed and accuracy now avail. Source list for I/O now incl.

Super Basic on Cassette **$40.00**

Tom Pittman's 1802 Tiny Basic Source listing now available. Find out how Tom Pittman wrote Tiny Basic and how to get the most out of it. Never offered before. **$19.00.**

S-100 4-Slot Expansion **$ 9.95**

Super Monitor VI.I Source Listing **$15.00**

Coming Soon: Assembler, Editor, Disassembler, DA/AD, Super Sound/Music, EPROM programmer, Stringy Floppy System.

RCA Cosmac Super Elf Computer $106.95

Compare features before you decide to buy any other computer. There is no other computer on the market today that has all the desirable benefits of the **Super Elf** for so little money. The Super Elf is a small single board computer that does many **big** things. It is an excellent computer for training and for learning programming with its machine language and yet it is easily **expanded with additional memory, Full Basic, ASCII Keyboards, video character generation, etc.**

Before you buy another small computer, see if it includes the following features: ROM monitor; State and Mode displays; Single step; Optional address displays; Power Supply; Audio Amplifier and Speaker; Fully socketed for all IC's; Real cost of in warranty repairs; Full documentation.

The **Super Elf** includes a **ROM** monitor for program loading, editing and execution with **SINGLE STEP for program debugging** which is not included in others at the same price. With **SINGLE STEP** you can see the microprocessor chip operating with the **unique Quest** address and data bus displays **before, during** and **after** executing instructions. Also, CPU mode and instruction cycle are decoded and displayed on 8 LED indicators.

An RCA 1861 **video graphics** chip allows you to connect to your own TV with an inexpensive video modulator to do graphics and games. There is a **speaker system** included for writing your own music or using many music programs already written. The speaker amplifier may also be used to drive relays for control purposes.

A 24 key **HEX keyboard** includes 16 HEX keys plus **load, reset, run, wait, input, memory protect, monitor select** and **single step**. Large, on board displays provide output and optional **high** and **low address**. There is a 44 pin standard connector slot for PC cards and a 50 pin connector slot for the Quest Super Expansion Board. Power supply and sockets for **all** IC's are included in the price plus a detailed 127 pg. instruction manual which now includes over 40 pgs. of software info. including a series of lessons to help get you started and a music program and graphics target game. Many schools and universities are using the Super Elf as a course of study. OEM's use it for training and R&D.

Remember, other computers only offer Super Elf features at additional cost or not at all. **Compare before you buy. Super Elf Kit $106.95, High address option $8.95, Low address option $9.95. Custom Cabinet** with drilled and labelled plexiglass front panel **$24.95.** Expansion Cabinet with room for 4 S-100 boards **$41.00. NiCad Battery Memory Saver Kit $6.95.** All kits and options also completely assembled and tested.

Questdata, a 12 page monthly software publication for 1802 computer users is available by subscription for $12.00 per year. Issues 1-12 bound $16.50.

Tiny Basic Cassette **$10.00**, on ROM **$38.00**, original Elf kit board **$14.95. 1802 software;** Moews Video Graphics **$3.50.** Games and Music **$3.00**, Chip 8 Interpreter **$5.50.**

Super Expansion Board with Cassette Interface $89.95

This is truly an astounding value! This board has been designed to allow **you** to decide how you want it optioned. **The Super Expansion Board comes with 4K of low power RAM** fully addressable anywhere in 64K with built-in memory protect and a **cassette interface**. Provisions have been made for all other options on the same board and it fits neatly into the hardwood cabinet alongside the **Super Elf**. The board includes slots for up to 6K of **EPROM** (2708, 2758, 2716 or TI 2716) and is **fully socketed.** EPROM can be used for the monitor and Tiny Basic or other purposes.

A 1K Super ROM Monitor $19.95 is available as an on board option in 2708 EPROM which has been preprogrammed with a program loader/editor and error checking multi file cassette read/write software, (relocatable cassette file) another exclusive from Quest. It includes register save and readout, block move capability and video graphics driver with blinking cursor. Break points can be used with the register save feature to isolate program bugs quickly, then follow with single step. The Super Monitor is written with subroutines allowing users to take advantage of monitor functions simply by calling them up. Improvements and revisions are easily done with the monitor. If you have the **Super Expansion Board** and **Super Monitor** the monitor is up and running at the push of a button.

Other on board options include **Parallel Input and Output Ports with full handshake.** They allow easy connection of an ASCII keyboard to the input port. **RS 232 and 20 ma Current Loop** for teletype or other device are on board and if you need more memory there are two **S-100** slots for static RAM or video boards. Also a 1K Super Monitor version 2 with video driver for full capability display with Tiny Basic and a video interface board. **Parallel I/O Ports $9.85, RS 232 $4.50, TTY 20 ma I/F $1.95, S-100 $4.50. A 50 pin connector set** with ribbon cable is available at $15.25 for easy connection between the **Super Elf** and the **Super Expansion Board.**

Power Supply Kit for the complete system (see Multi-volt Power Supply below).

Same day shipment. First line parts only. Factory tested. Guaranteed money back. Quality IC's and other components at factory prices.

INTEGRATED CIRCUITS

7400TTL		LM323K-5	5.95	CD4026	2.50
7400N	.19	LM320K-12	1.50	CD4027	.66
7402N	.20	LM320K-15	1.50	CD4028	.85
7404N	.25	LM320T-5	1.35	CD4029	1.35
7409N	.23	LM320T-8	1.35	CD4030	.45
7410N	.20	LM320T-12	1.35	CD4035	1.35
7414N	.70	LM320T-15	1.35	CD4040	1.35
7420N	.20	LM324N	1.40	CD4042	.85
7422N	.39	LM339N	1.00	CD4043	.85
7430N	.20	LM340K-5	1.35	CD4044	.85
7442N	.58	LM340K-8	1.35	CD4046	1.67
7445N	.74	LM340K-12	1.35	CD4049	.45
7447N	.58	LM340K-15	1.35	CD4050	.49
7448N	.77	LM340K-24	1.35	CD4051	1.13
7450N	.20	LM340T-5	1.25	CD4060	1.42
7474N	.35	LM340T-8	1.25	CD4066	.71
7475N	.49	LM340T-12	1.25	CD4068	.40
7485N	.88	LM340T-15	1.25	CD4069	.40
7489N	1.85	LM340T-18	1.25	CD4070	.50
7490N	.43	LM340T-24	1.25	CD4071	.45
7492N	.43	LM350	7.50	CD4072	.45
7493N	.43	LM377	3.50	CD4073	.45
7495N	.69	LM379	5.00	CD4075	.45
74100N	1.20	LM380N	1.00	CD4076	1.65
74107N	.35	LM381	1.60	CD4078	.40
74121N	.34	LM382	1.60	CD4081	.35
74123N	.59	LM703H	.65	CD4082	.35
74125N	.45	LM709H	.28	CD4116	.47
74145N	.77	LM723H/N	.50	CD4490	5.50
74150N	1.20	LM733N	.85	CD4507	1.00
74151N	.69	LM741CH	.35	CD4508	4.25
74154N	2.00	LM741N	.38	CD4510	1.02
74157N	.69	LM747H/N	.75	CD4511	.94
74161N	.87	LM748N	.35	CD4515	2.52
74162N	1.25	LM1303N	1.75	CD4516	1.10
74163N	.87	LM1304	1.10	CD4518	1.02
74174N	.96	LM1305	1.27	CD4520	1.02
74175N	.95	LM1307	2.00	CD4527	1.51
74190N	1.15	LM1310	2.75	CD4528	.79
74192N	.87	LM1458	.47	CD4553	3.50
74193N	.85	LM1800	1.75	CD4566	2.25
74221N	2.75	LM1812	7.50	CD4583	2.35
74298N	1.65	LM1889	3.00	CD4585	1.10
74365N	.89	LM2111	1.75	CD40192	3.00
74366N	.89	LM2902	2.25	74C00	.35
74367N	.89	LM3900N	.60	74C04	.40
		LM3905	1.75	74C10	.35
74LS00 TTL		LM3909N	.95	74C14	1.95
74LS00N	.35	MC1458V	.50	74C20	.35
74LS02N	.35	NE550N	1.00	74C30	.35
74LS04N	.55	NE555V	.39	74C48	2.25
74LS05N	.40	NE556A	.85	74C74	.85
74LS08N	.45	NE565A	1.00	74C76	1.75
74LS10N	.45	NE566V	1.50	74C90	1.75
74LS13N	.55	NE567V	1.00	74C93	1.75
74LS14N	1.25	NE570B	4.75	74C154	3.00
74LS20N	.45	78L05	.60	74C160	2.00
74LS22N	.45	78L08	.60	74C175	1.35
74LS28N	.45	78M05	.85	74C192	2.25
74LS30N	.45	75108	1.75	74C221	2.50
74LS33N	.75	75491CN	.50	74C905	6.00
74LS38N	.75	75492CN	.55	74C906	.75
74LS74N	1.25	75494CN	.89	74C914	1.95
74LS75N	1.00			74C922	6.00
74LS90N	1.00	**A to D CONVERTER**		74C923	6.00
74LS93N	1.00	8038B	4.50	74C925	7.50
74LS95N	1.10	8700CJ	13.95	74C926	6.95
74LS107N	.65	8701CN	22.00	74C927	6.95
74LS112N	.65	8750CJ	13.95		
74LS113N	.45	LD130	9.95	**INTERFACE**	
74LS132N	.89	9400CJV/F	7.40	8095	.65
74LS136N	.65	ICL7103	9.50	8096	.65
74LS151N	1.10	ICL7107	14.25	8097	.65
74LS155N	1.10			8098	.65
74LS157N	1.10	**CMOS**		8T09	1.25
74LS162N	1.65	CD4000	.25	8T10	4.50
74LS163N	1.65	CD4001	.35	8T13	3.00
74LS174N	2.00	CD4002	.35	8T20	5.50
74LS190N	1.25	CD4006	1.10	8T23	3.10
74LS221N	2.50	CD4007	.35	8T24	3.50
74LS258N	1.60	CD4008	.28	8T25	3.20
74LS367N	1.65	CD4009	.45	8T26	1.69
		CD4010	.45	8T28	2.75
LINEAR		CD4011	.35	8T97	1.69
CA3045	.90	CD4012	.28	8T98	1.69
CA3046	1.10	CD4013	.47		
CA3081	1.80	CD4014	1.25	**MOS/MEMORY RAM**	
CA3082	1.90	CD4015	1.00	2101-1	2.95
CA3089	2.95	CD4016	.55	2102-1	.95
LM301AN/AH	.35	CD4017	1.05	2102AL-4	1.25
LM305H	.87	CD4018	.94	2102AN-2L	1.60
LM307N	.35	CD4019	.45	21L02-1	1.18
LM308N	1.00	CD4020	1.02	2104A-4	4.95
LM309K	1.50	CD4021	1.35	2107B-4	3.75
LM311H/N	.90	CD4022	1.10	2111-1	3.75
LM317T/K	3.75	CD4023	.28	2112-2	3.95
LM318	1.35	CD4024	.75	2114L 300ns	7.40
LM320K-5	1.50	CD4025	.28	2114L 450ns	6.40

4116-200ns	9.50
8/4116 200ns	60.00
2513B	6.30
MM5262	.40
MM5280	3.00
MM5320	9.95
MM5330	5.94
PD411D-3	4.00
PD411D-4	5.00
P5101L	8.95
4200A	9.95
82S25	2.90
91L02A	1.50
HD0165-5	6.95
MM57100	4.50
GIAY38500-1	9.95
MCM66751A	9.95
9368	3.50
410D	10.00
416	16.00
CLOCKS	
MM5311	5.50
MM5312	3.90
MM5314	3.90
MM5369	2.10
MM5841	14.45
MM5865	7.95
CT7001	5.80
CT7010	8.95
CT7015	8.95
MM5375AA/N	3.90
MM5375AG/N	4.90
7205	16.50
7207	7.50
7208	15.95
7209	4.95
DS0026CN	3.75
DS0056CN	3.75
MM53104	2.50
MICROPROCESSOR	
6502	10.95
6504	9.95
6522	9.95
6800	6.95
6802	11.95
6820	4.95
6850	5.95
8080A	5.95
8085	12.95
8086	75.00
Z80	9.95
Z80A	11.95
8212	2.90
8214	3.95
8216	2.90
8224	3.45
8228	4.95
8251	6.95
8253	15.00
8255	5.75
8257	10.95
8259	14.95
1802CP plas.	13.95
1802DP plas.	17.95
1861P	11.50
CDP1802CD	19.95
CDP1802D	25.00
CDP1861	12.95
UART/FIFO	
AY5-1013	5.50
AY5-1014	7.50
3341	6.95
PROM	
1702A	4.95
2513B upper case	6.95
2708	7.75
2716T1	24.50
2716 Intel	34.95
8/2716 Intel	240.00
2732	85.00
2758	22.50
8741A	85.00
8748	75.00
8748-8	70.00
8755A	65.00
N82S23	2.95
N82S123	6.50
N82S126	3.75
N82S129	8.50
N82S131	8.50
N82S136	8.75
N82S137	8.75
DM8577	2.90
8223	2.90

CONNECTORS

30 pin edge	2.50
44 pin edge	2.75
100 pin edge	4.50
100 pin edge WW	5.25

IC SOCKETS
Solder Tin Low Profile

PIN	1 UP	PIN	1UP
8	.15	22	.30
14	.14	24	.35
16	.16	28	.42
18	.27	36	.58
20	.29	40	.57

2 level 14 pin ww .20

WIRE WRAP LEVEL 3

PIN		PIN	
14	.32	24	.86
16	.33	28	1.00
18	.57	40	1.23

CRYSTALS

1 MHz	4.50
2 MHz	4.50
4 MHz	4.25
5 MHz	4.25
10 MHz	4.25
18 MHz	3.90
20 MHz	3.90
32 MHz	3.90
32768 Hz	4.00
1.8432 MHz	4.50
3.5795 MHz	1.20
2.0100 MHz	1.95
2.097152 MHz	4.50
2.4576 MHz	4.50
3.2768 MHz	4.50
5.0688 MHz	4.50
5.185 MHz	4.50
5.7143 MHz	4.50
6.5536 MHz	4.50
14.31818 MHz	4.25
18.432 MHz	4.50
22.1184 MHz	4.50

KEYBOARD ENCODERS

AY5-2376	$12.50
AY5-3600	17.95
AY5-9100	10.50
AY5-9200	16.50
74C922	5.50
74C923	5.50
HD0165-5	6.95
AY5-9400	10.50

D Connectors RS232

DB25P	2.95
DB25S	3.95
Cover	1.50
DE9S	1.95
DA15P	2.10
DA15S	3.10

Hickok 3½ Digit LED multimeter	**69.95**
Stopwatch Kit	**26.95**
Auto Clock Kit	**17.95**
Digital Clock Kit	**14.95**

8K/16K Eprom Kit (less PROMS)	$89.00
Motherboard	$39.00
Extender Board	$8.99

RESISTORS ¼ watt 5%
10 per type .03
25 per type .025
100 per type .015
1000 per type .012
350 piece pack
5 per type 6.75
½ watt 5% per type .05

Televideo Terminal 912 $845.00

KEYBOARDS

56 key ASCII keyboard kit	$67.50
Fully assembled	77.50
53 key ASCII keyboard kit	60.00
Fully assembled	70.00
Enclosure Plastic	14.95
Metal Enclosure	29.95

LEDS

Red T018	.15
Green, Yellow T018	.20
Jumbo Red	.20
Green, Orange, Yellow Jumbo	.25
Cliplite LED Mounting Clips	8/$1.25

(specify red, amber, green, yellow, clear)

CONTINENTAL SPECIALTIES in stock
Complete line of breadboard test equip.
MAX-100 8 digit Freq. Ctr. $128.95

OK WIRE WRAP TOOLS in stock
Portable Multimeter $18.00

Complete line of AP Products in stock.

SPECIAL PRODUCTS

MM5865 Stopwatch Timer with 10 pg. spec.	9.00
PC board	7.50
Switches Mom. Pushbutton	.27
3 pos. slide	.25
Encoder HD0165-5	6.95
Paratronics 100A Logic Analyzer Kit	**$224.00**
Model 10 Trigger Expander Kit	$229.00
Model 150 Bus Grabber Kit	$369.00
Sinclair 3½ Digit Multimeter	**$59.95**
Clock Calendar Kit	**$23.95**
2.5 MHz Frequency Counter Kit	**$37.50**
30 MHz Frequency Counter Kit	**$47.75**

TRANSFORMERS

6V 300 ma	3.25
12 Volt 300 ma transformer	1.25
12.6V CT 600 ma	3.75
12V 250 ma wall plug	2.95
12V CT 250 ma wall plug	3.50
24V CT 400 ma	3.95
10V 1.2 amp wall plug	4.85
12V 6 amp	12.95
12V 500 ma wall plug	4.75
12V 1 amp wall plug	6.50
12V 3 amp	8.50
10/15 VAC 8/16 VA wall plug	9.75

DISPLAY LEDS

MAN1	CA	.270	2.90
MAN3	CC	.125	.39
MAN72/74	CA/CA	.300	1.00
DL704	CC	.300	1.25
DL707/DL707R	CA	.300	1.00
DL727/728	CA/CC	.500	1.90
DL747/750	CA/CC	.600	1.95
DL750	CC	.600	1.95
FND359	CC	.357	.70
FND500/507	CC/CA	.500	1.35
FND503/510	CC/CA	.500	.90
FND800/807	CC/CA	.800	2.20
3 digit Bubble			.60
4 digit Bubble			.80
DG8 Fluorescent			1.75
DG10 Fluorescent			1.75
5 digit 14 pin display			1.00
10 digit display			1.25
7520 Clairex photocells			.39
TIL311 Hex			9.50
MAN3640	CC	.30	1.10
MAN4610	CA	.40	1.20
MAN4640	CC	.40	1.20
MAN4710	CA	.40	.95
MAN4740	CC	.40	1.20
MAN6640	CC	.56	2.95
MAN6710	CA	.60	1.35
MAN6740	CC	.60	1.35

MA1002A	**8.95**
MA1002E	**8.95**
MA1012A	**8.95**
102P3 transformer	**2.25**
MA1012A Transformer	**2.25**

Rockwell AIM 65 Computer

6502 based single board with full ASCII keyboard and 20 column thermal printer. 20 char. alphanumeric display, ROM monitor, fully expandable. **$375.00.** 4K version **$450.00.** 4K Assembler **$85.00**, 8K Basic Interpreter **$100.00.**

Special small power supply for AIM65 assem. in frame **$49.00.** Complete AIM65 in thin briefcase with power supply **$485.00.** Molded plastic enclosure to fit AIM65 plus power supply **$47.50.** Special Package Price: 4K AIM, 8K Basic, power supply, cabinet **$599.00**

AIM65/KIM/VIM/Super Elf 44 pin expansion board; 3 female and 1 male bus. Board plus 3 connectors **$22.95.**

AIM65/KIM/VIM I/O Expansion Kit; 4 parallel and 2 serial ports plus 2 internal timers **$39.00.** PROM programmer for 2716 **$150.00.** 32K RAM Board assem. **$419.00.** 16K RAM assem. **$360.00**

Multi-volt Computer Power Supply

8v 5 amp, ±18v .5 amp, 5v 1.5 amp, −5v .5 amp, 12v .5 amp, −12 option. ±5v, ±12v are regulated. Kit $29.95. Kit with punched frame $37.45, $4.00 shipping. Kit of hardware $14.00. Woodgrain case $10.00, $1.50 shipping.

PROM Eraser

Will erase 25 PROMs in 15 minutes. Ultraviolet, assembled **$37.50**
Safety switch/Timer version **$69.50**

60 Hz Crystal Time Base Kit $4.40

Converts digital clocks from AC line frequency to crystal time base. Outstanding accuracy.

NiCad Battery Fixer/Charger Kit

Opens shorted cells that won't hold a charge and then charges them up, all in one kit w/full parts and instructions. **$7.25**

LRC 7000+Printer $389.00

40/64 column dot matrix impact, std. paper. Interface all personal computers.

S-100 Computer Boards

8K Static RAM Kit	$129.00
8K Static Godbout Econo IIA Kit	145.00
16K Static Godbout Econo XIV Kit	$285.00
24K Static Godbout Econo VIIA-24 Kit	435.00
32K Static Godbout Econo X-32 Kit	$575.00
16K Dynamic RAM Kit	199.00
32K Dynamic RAM Kit	310.00
64K Dynamic RAM Kit	470.00
Video Interface Kit	**$129.00**

80 IC Update Master Manual $55.00

Complete IC data selector, 2700 pg. master reference guide. Over 51,000 cross references. Free update service through 1980. Domestic postage $3.50. 79IC Master closeout $29.95.

Z80 Microcomputer

16 bit I/O, 2 MHz clock, 2K RAM, ROM Breadboard space. Excellent for control. Base Board $28.50. Full Kit $99.00. Monitor $20.00. Power Supply Kit $35.00.

Video Modulator Kit $8.95

Convert TV set into a high quality monitor w/o affecting usage. Comp. kit w/full instruc.

Modem Kit $60.00

State of the art, orig., answer. No tuning necessary. 103 compatible 300 baud. Inexpensive acoustic coupler plans included.

BSR Controller $39.95

Connect your computer to the BSR Home Control System. Computer controlled ultrasonic transmitter for your BSR. Software for 1802 user.

TERMS: $5.00 min. order U.S. Funds. Calif residents add 6% tax. BankAmericard and Master Charge accepted. Shipping charges will be added on charge cards.

FREE: Send for your copy of our NEW 1980 QUEST CATALOG. Include 28¢ stamp.

DEALER DIRECTORY

Los Angeles, CA
Featuring: PolyMorphic, North Star, Imsai, Cromemco, Extensys, Speechlab products and Poly-88 Users Group software exchange. All products 10-20% off list. We won't be undersold! **A-A-A-A Discount Computer How's, 1477 Barrington, Suite 17, Los Angeles, CA 90025, 477-8478.**

San Jose, CA
Bay area's newest computer store. Featuring the new Texas Instruments TI 99/4 home & business computer. Software for TRS-80, Apple, PET, etc. Magazines. **Hobbi-Tronics, 1378 S. Bascom Ave., San Jose, CA 95128, 998-1103.**

Sarasota, FL
Dynabyte computer systems, Hazeltine and NEC, Word-Star, Structured Systems accounting. Consulting, training, sales, service. **Glisco, Inc., 4001 Roberts Point Rd., Sarasota, FL 33581, 349-0200.**

Venice, FL
Discount prices & professional service: Cromemco, North Star, Vector Graphic, DEC, TI, Thinker Toys, Intertube, Soroc, Centronics, NEC, Selectric interfaces, Microdasys. Complete business & medical billing software available. MicroAge & Serendipity software discounted. **Sara-Tech Electronics, Inc., Computer Division, PO Box 692, Venice, FL 33595, 485-3559.**

Say you saw thier ad in Microcomputing

Aurora, IL
Microcomputer systems for home or business; peripherals, software, books & magazines. Apple, Hewlett-Packard, North Star, Cromemco systems. IDS-440G printer w/Apple graphics, New HP-85 & HP calculators. **Farnsworth Computer Center, 1891 N. Farnsworth Ave., Aurora, IL 60505, 851-3888.**

Chicago, IL
Computer Hardware/Software Specialists for home and business. Largest selection of computer books, magazines and copyrighted software in Chicago Metro area. Experienced factory-trained service department. Feature Apple and Alpha Microsystems and accessories. **Data Domain of Schaumburg, 1612 E. Algonquin Road., Schaumburg, IL 60195, 397-8700.**

Naperville, IL
Computer systems design, programming and consultation by computer experts. Dealer for SSM, Integrand, Tarbell, Ithaca Intersystems, Verbatim, Diablo and others. Discount prices on many items. **Wilcox Enterprises, 25W178-39th St., Naperville, IL 60540, 420-8601.**

Laurel, MD
Apple & Exidy Sorcerer. Authorized dealer for both. We also sell subscriptions to "The Source" timeshare service. Plus full line of ham equipment. **The Comm Center, Laurel Plaza, Rte. 198, Laurel, MD 20810, 792-0600.**

Worcester, MA
Complete business and personal systems on Data General, microNova and Apple. Software, books, magazines, accessories and supplies. Custom business programming available. Authorized Apple service. Downtown. **The Computer Place Inc., 11 Harvard St., Worcester, MA 01609, 755-5387.**

Garden City, MI
Complete systems for business, professional and personal applications. Custom programming available. Apple II, North Star, Vector Graphic and other lines of microcomputers, software, books, components. **Computer Center, 28251 Ford Rd., Garden City, MI 48135, 422-2570.**

St. Louis, MO
Experimenters' Paradise. Electronic and mechanical components. Computer People, Audio People, Hams, Robot Builders, Experimenters. Open six days a week. **Gateway Electronics Corp., 8123-25 Page Blvd., St. Louis, MO 63130, 427-6116.**

Akron, OH
We've got it all. Business systems. Personal systems. Software packages. Custom programming. Terminals. Printers. Service and books. Easy freeway access. 10 AM to 6 PM Monday–Saturday. **The Basic Computer Shop, Fairlawn Plaza, 2671 West Market St., Akron, OH 44313, 867-0808.**

Portland, OR
Ohio Scientific specialists for business and personal computers. Local service. Terminals, printers, custom programming. Full OSI product line on display! 10AM to 6PM M–F. **Fial Computer, 11266 SE 21st Ave., Milwaukie, OR 97222, 654-9574.**

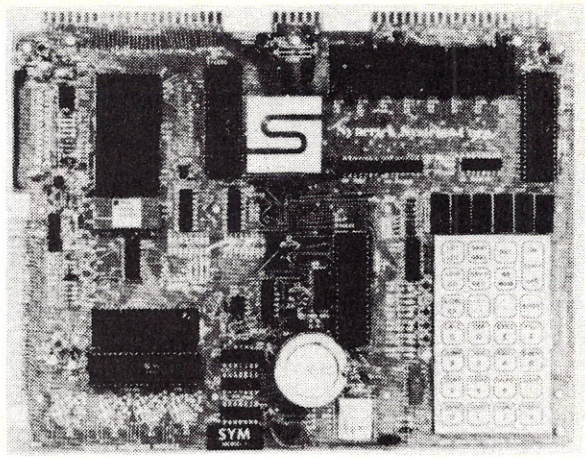
Kingston, PA
We support Level II and Model II. Books, magazines, programs, parts, accessories, peripherals, free literature, free seminars, cassettes, floppies, filters, transformers, caps, chips, CRTs. **Artco Electronics, 302 Wyoming Ave., Kingston, PA 18704, 287-1014.**

Sara, Mexico
Learn how to utilize and program the Z-80 microprocessor to maintain your company's records in top shape. This technique has just been introduced to Mexico. Courses, applications, maintenance, service. **Digitales, S.A. de C.V. Sara 4612, Mexico 14, D.F., 5-17-41-59.**

Dealers: Listings are **$15** per month in prepaid quarterly payments, or one yearly payment of **$150**, also prepaid. Ads include 25 words describing your products and services plus your company name, address and phone. (No area codes or merchandise prices, please.) Call Marcia at 603-924-7138 or write *Kilobaud Microcomputing*, Ad Department, Peterborough, NH 03458.

SYM-1, 6502-BASED MICROCOMPUTER

- FULLY-ASSEMBLED AND COMPLETELY INTEGRATED SYSTEM that's ready-to-use
- ALL LSI IC'S ARE IN SOCKETS
- 28 DOUBLE-FUNCTION KEYPAD INCLUDING UP TO 24 "SPECIAL" FUNCTIONS
- EASY-TO-VIEW 6-DIGIT HEX LED DISPLAY
- KIM-1* HARDWARE COMPATIBILITY
  The powerful 6502 8-Bit MICROPROCESSOR whose advanced architectural features have made it one of the largest selling "micros" on the market today.
- THREE ON-BOARD PROGRAMMABLE INTERVAL TIMERS available to the user, expandable to five on-board.
- 4K BYTE ROM RESIDENT MONITOR and Operating Programs.
- Single 5 Volt power supply is all that is required.
- 1K BYTES OF 2114 STATIC RAM onboard with sockets provided for immediate expansion to 4K bytes onboard, with total memory expansion to 65, 536 bytes.
- USER PROM/ROM: The system is equipped with 3 PROM/ROM expansion sockets for 2316/2332 ROMs or 2716 EPROMs
- ENHANCED SOFTWARE with simplified user interface
- STANDARD INTERFACES INCLUDE:
  —Audio Cassette Recorder Interface with Remote Control (Two modes: 135 Baud KIM-1* compatible, Hi-Speed 1500 Baud)
  —Full duplex 20mA Teletype Interface
  —System Expansion Bus Interface
  —TV Controller Board Interface
  —CRT Compatible Interface (RS-232)
- APPLICATION PORT: 15 Bi-directional TTL Lines for user applications with expansion capability for added lines
- EXPANSION PORT FOR ADD-ON MODULES (51 I/O Lines included in the basic system)
- SEPARATE POWER SUPPLY connector for easy disconnect of the d-c power
- AUDIBLE RESPONSE KEYPAD



Synertek has enhanced KIM-1* software as well as the hardware. The software has simplified the user interface. The basic SYM-1 system is programmed in machine language. Monitor status is easily accessible, and the monitor gives the keypad user the same full functional capability of the TTY user. The SYM-1 has everything the KIM-1* has to offer, plus so much more that we cannot begin to tell you here. So, if you want to know more, the SYM-1 User Manual is available, separately.

SYM-1 Complete w/manuals	**$229.00**
SYM-1 User Manual Only	**$7.00**
SYM-1 Expansion	**$60.00**

Expansion includes 3K of 2114 RAM chips and 1-6522 I/O chip.

SYM-1 Manuals: The well organized documentation package is complete and easy-to-understand.

SYM-1 CAN GROW AS YOU GROW. It's the system to BUILD-ON.
Expansion features that are available:

BAS-1 8K Basic ROM (Microsoft Basic)	**$89.00**
KTM-2 (Complete terminal less monitor)	**$319.00**

QUALITY EXPANSION BOARDS DESIGNED SPECIFICALLY FOR KIM-1, SYM-1 & AIM 65

These boards are set up for use with a regulated power supply such as the one below, but, provisions have been made so that you can add onboard regulators for use with an unregulated power supply. But, because of unreliability, we do not recommend the use of onboard regulators. All I.C.'s are socketed for ease of maintenance. *All boards carry full 90-day warranty.*

All products that we manufacture are designed to meet or exceed industrial standards. All components are first qualtiy and meet full manufacturer's specifications. All this and an extended burn-in is done to reduce the normal percentage of field failures by up to 75%. To you, this means the chance of inconvenience and lost time due to a failure is very rare; but, if it should happen, we guarantee a turn-around time of less than forty-eight hours for repair.

Our money back guarantee: If, for any reason you wish to return any board that you have purchased directly from us within ten (10) days after receipt, complete, in original condition, and in original shipping carton; we will give you a complete credit or refund less a $10.00 restocking charge per board.

VAK-1 8-SLOT MOTHERBOARD

This motherboard uses the KIM-4* bus structure. It provides eight (8) expansion board sockets with rigid card cage. Separate jacks for audio cassette, TTY and power supply are provided. Fully buffered bus.

VAK-1 Motherboard **$139.00**

VAK-2/4 16K STATIC RAM BOARD

This board using 2114 RAMs is configured in two (2) separately addressable 8K blocks with individual write-protect switches.

VAK-2 16K RAM Board with only 8K of RAM (½ populated)	**$239.00**
VAK-3 Complete set of chips to expand above board to 16K	**$125.00**
VAK-4 Fully populated 16K RAM	**$325.00**

VAK-5 2708 EPROM PROGRAMMER

This board requires a +5 VDC and ±12 VDC, but has a DC to DC multiplyer so there is no need for an additional power supply. All software is resident in on-board ROM, and has a zero-insertion socket.

VAK-5 2708 EPROM Programmer **$249.00**

VAK-6 EPROM BOARD

This board will hold 8K of 2708 or 2758, or 16K of 2716 or 2516 EPROMs. *EPROMs not included.*

VAK-6 EPROM Board **$119.00**

VAK-7 COMPLETE FLOPPY-DISK SYSTEM **$1299.00**

See May Kilobaud for details

VAK-8 PROTYPING BOARD

This board allows you to create your own interfaces to plug into the motherboard. Etched circuitry is provided for regulators, address and data bus drivers; with a large area for either wire-wrapped or soldered IC circuitry.

VAK-8 Protyping Board **$39.00**

POWER SUPPLIES

ALL POWER SUPPLIES are totally enclosed with grounded enclosures for safety, AC power cord, and carry a full 2-year warranty.

FULL SYSTEM POWER SUPPLY

This power supply will handle a microcomputer and up to 65K of our VAK-4 RAM. ADDITIONAL FEATURES ARE: Over voltage Protection on 5 volts, fused, AC on/off switch. Equivalent to units selling for $225.00 or more.

Provides +5 VDC @ 10 Amps & ±12 VDC @ 1 Amp
VAK-EPS Power Supply **$119.00**

VAK-EPS/AIM — same as VAK-EPS but w/additional 24 volt unregulated (specifically for AIM 65) **$149.00**

KIM-1* Custom P.S. provides 5 VDC @ 1.2 Amps and +12 VDC @ .1 Amps
KCP-1 Power Supply **$39.00**

*KIM is a product of MOS Technology

Add $2.50 for shipping & handling for all <u>except</u> AIM 65.

RNB ENTERPRISES
INCORPORATED

2967 W. Fairmount Avenue
Phoenix AZ. 85017
(602)265-7564

VISA

master charge

IT'S KID JADE'S SURPLUS SALE.

S-100 Boards

THE BIG Z - Jade
2 or 4 MHz switchable Z-80 CPU with serial I/O

CPU-30201K	*Kit*	$135.00
CPU-30201A	*A & T*	$199.00
CPU-30200B	*Bare board*	$35.00

SBC-100 - SD Systems
2.5 MHz Z-80 CPU with serial & parallel I/O ports

CPC-30100K	*Kit*	$257.50
CPC-30100A	*A & T*	$325.00

SBC-200 - SD Systems
4 MHz Z-80 CPU with serial & parallel I/O ports

CPC-30200K	*Kit*	$399.95
CPC-30200A	*A & T*	$475.00

CB2 - S.S.M.
2 or 4 MHz switchable Z-80 CPU with RAM, ROM, & I/O

CPU-30300K	*Kit*	$185.00
CPC-30300A	*A & T*	$249.95

ExpandoRAM I - SD Systems
2.5 MHz RAM board expandable from 16K to 64K

MEM-16130K	*16K kit*	$199.95
MEM-16130A	*16K A & T*	$249.95
MEM-32131K	*32K kit*	$254.95
MEM-32131A	*32K A & T*	$304.95
MEM-48132K	*48K kit*	$309.95
MEM-48132A	*48K A & T*	$359.95
MEM-64133K	*64K kit*	$364.95
MEM-64133A	*64K A & T*	$414.95

ExpandoRAM II - SD Systems
4 MHz RAM board expandable from 16K to 256K

MEM-16630A	*16K kit*	$249.95
MEM-16630A	*16K A & T*	$299.95
MEM-32631K	*32K kit*	$309.95
MEM-32631A	*32K A & T*	$359.95
MEM-48632K	*48K kit*	$369.95
MEM-48631A	*48K A & T*	$409.95
MEM-64633K	*64K kit*	$429.95
MEM-64633A	*64K A & T*	$479.95

32K STATIC RAM - Jade
2 or 4 MHz expandable static RAM board uses 2114L's

MEM-16150K	*16K 2 MHz kit*	$249.95
MEM-16150A	*16K 2 MHz A & T*	$299.95
MEM-16151K	*16K 4 MHz kit*	$259.95
MEM-16151A	*16K 4 MHz A & T*	$309.95
MEM-32150K	*32K 2 MHz kit*	$399.95
MEM-32150A	*32K 2 MHz A & T*	$449.95
MEM-32151K	*32K 4 MHz kit*	$409.95
MEM-32151A	*32K 4 MHz A & T*	$459.95

16K STATIC RAM - Cal Comp Sys
2 or 4 MHz 16K static RAM - a real memory bargain

MEM-16160K	*16K 2 MHz kit*	$249.95
MEM-16160A	*16K 2 MHz A & T*	$279.00
MEM-16162K	*16K 4 MHz kit*	$279.95
MEM-16162A	*16K 4 MHz A & T*	$309.00
MEM-16160B	*Bare board*	$29.95

PB-1 - S.S.M.
2708, 2716 EPROM board with built-in programmer

MEM-99510K	*Kit*	$119.95
MEM-99510A	*A & T*	$169.95

PROM-100 - SD Systems
2708, 2716, 2732, 2758, & 2516 EPROM programmer

MEM-99520K	*Kit*	$175.00
MEM-99520A	*A & T*	$225.00

32K BYTESAVER - Cromemco
32K EPROM board with on-board 2716 programmer

MEM-32550A	*A & T*	$295.00

Call for your free 1980 catalog

DOUBLE-D - Jade
Double density controller with the inside track

IOD-1200K	*Kit*	$299.95
IOD-1200A	*8" A & T*	$389.95
IOD-1205A	*5¼" A & T*	$389.95
IOD-1200B	*Bare board*	$65.00

VERSAFLOPPY I - SD Systems
Versatile floppy disk controller for 8" or 5¼"

IOD-1150K	*Kit*	$219.95
IOD-1150A	*A & T*	$269.95

VERSAFLOPPY II - SD Systems
New double density controller for both 8" & 5¼"

IOD-1160K	*Kit*	$309.95
IOD-1160A	*A & T*	$369.95

I/O-4 - S.S.M.
2 serial I/O ports plus 2 parallel I/O ports

IOI-1010K	*Kit*	$129.95
IOI-1010A	*A & T*	$189.95
IOI-1010B	*Bare board*	$29.95

BIT STREAMER II - Vector Graphic
3 serial I/O ports plus 2 parallel I/O ports

IOI-1025A	*A & T*	$259.00

CB-1 - Mullen
S-100 controller for up to 8 external devices

IOC-1100K	*Kit*	$119.95
IOC-1100A	*A & T*	$169.95

100K DAY CLOCK - Mtn Hardware
Crystal controlled S-100 clock with NiCad backup

IOK-1400A	*A & T*	$329.95

SB1 - S.S.M.
15 Hz to 25K Hz music synthesizer for S-100

IOS-1005K	*Kit*	$189.95
IOS-1005A	*A & T*	$269.95

TB-4 - Mullen
Extremely versatile extender board with logic probe

TSX-180K	*Kit*	$55.00
TSX-180A	*A & T*	$75.00

TERMINATOR & EXTENDER - C.C.S.
Can be used as both an S-100 extender and terminator

TSX-150K	*Kit*	$39.95

S-100 EXTENDER - Cal Comp Sys
Puts problem boards within easy reach

TSX-160A	*A & T*	$24.95

VDB-8024 - SD Systems
80 x 24 I/O mapped video board with keyboard I/O

IOV-1020K	*Kit*	$324.95
IOV-1020A	*A & T*	$379.95

VB3 - S.S.M.
80 x 24 or 80 x 51 memory mapped with graphics

IOV-1090K	*Kit, 2 MHz*	$299.95
IOV-1090A	*A & T, 2 MHz*	$385.00
IOV-1095K	*Kit, 4 MHz*	$339.95
IOV-1095A	*A & T, 4 MHz*	$399.00
IOV-1091K	*80 x 51 upgrade, 2 MHz*	$69.00
IOV-1096K	*80 x 51 upgrade, 4 MHz*	$89.00

VIDEO BOARD - Jade
64 x 16 assembled & tested S-100 video board

IOV-1050A	*A & T sale price*	$135.00

Modems

NOVATION CAT
300 baud, auto answer/originate acoustic modem

IOM-5200A	*Special sale price*	$149.95

Call for your free 1980 catalog

Single Board Computer

AIM-65 - Rockwell
6502 computer with printer, display, & keyboard

CPK-50165	*1K AIM*	$374.95
CPK-50465	*4K AIM*	$449.95
SFK-74600008E	*8K BASIC ROM*	$99.95
SFK-64600004E	*4K assembler ROM*	$84.95
PSX-030A	*Power supply*	$59.95
ENX-000002	*Enclosure*	$49.95

4K AIM, 8K BASIC, power supply, & enclosure

Special package price $599.00

32K RAM - for AIM-65
Dynamic memory board to expand your AIM-65

MEM-99170A	*A & T w/out RAM*	$275.00
MEM-16170A	*A & T w/16K*	$325.00
MEM-32170A	*A & T w/32K*	$375.00
MEM-99170B	*Bare board*	$49.00

DISK CONTROLLER - for AIM-65
Add 5¼" or 8" disk drives to your AIM-65

IOD-3013A	*A & T*	$575.00

VISIBLE MEMORY - for AIM-65
Video board with 8K memory & graphics for AIM-65

IOV-3011A	*A & T*	$239.95

MEMORY-MATE - for AIM-65
The master-mater with 48K RAM, I/O, PROM, & music

MEM-52301A	*A & T w/16K*	$475.00

Z-80 STARTER KIT - SD Systems
Z-80 computer with RAM, ROM, I/O, & keyboard

CPS-30010K	*Kit*	$289.95
CPS-30010A	*A & T*	$349.95

Accessories for Apple

16K MEMORY UPGRADE
Add 16K of RAM to your TRS-80, Apple, or Exidy

MEX-16100K	*TRS-80 kit*	$49.95
MEX-16101K	*Apple kit*	$49.95
MEX-16102K	*Exidy kit*	$49.95

DISK DRIVE for APPLE
5¼" disk drive with controller for your Apple

MSM-12310C	*with controller*	$495.00
MSM-123101	*w/out controller*	$425.00

8" DRIVES for APPLE
Controller, DOS, two 8" drives, cabinet, & cable

Special package price $1475.00

AIO - S.S.M.
Parallel & serial interface for your Apple

IOI-2050K	*Kit*	$115.00
IOI-2050A	*A & T*	$155.00

SUP'R'TERMINAL - M & R Assoc
80 x 24 video display board for your Apple

IOV-2100A	*A & T*	$359.00

SUPERTALKER - Mtn Hardware
Speech recognition/synthesizer w/speaker & mike

IOS-2015A	*A & T*	$275.00

Z-80 CARD for APPLE
Z-80 CPU card with CP/M for your Apple

CPX-30800A	*A & T*	$345.00

MICROMODEM - D.C. Hayes
Auto answer/dial modem card for Apple or S-100

IOM-2010A	*Apple modem*	$349.95
IOM-1100A	*S-100 modem*	$375.00

SUP'R'MOD II - M & R Assoc
Color or B & W TV interface recommended for Apple

IOR-5050A	*A & T*	$29.95

SUMMER '80 ROUND #4.

KNOCKOUT PRICES

Printers

BASE 2 - Impact Printer
132 cps, bi-directional, tractor feed, & graphics
PRM-13100 .... **$625.00**

DP-9500 - Anadex
9 x 9 dot matrix, 176 column, 200 cps, & graphics
PRM-10500 *Standard DP-9500* **$1395.00**
PRM-10510 *with graphics & 2K* .. **$1495.00**

LP-80 - Matchless
9 x 7 matrix, 132 column, 125 cps, bi-directional
PRM-37204 .... **$775.00**

PAPER TIGER - Integral Data
132 column, parallel & serial, 150 cps, graphics
PRM-33440 *IDS-440* **$950.00**
PRM-33441 *IDS-440 w/graphics* .. **$1050.00**

MIPLOT - Watanabe Instruments
Intelligent graphics plotter uses 7 bit ASCII code
PRP-10800 .... **$1075.00**

SPINWRITER - NEC
65 cps, bi-directional, letter quality with tractor
PRD-55510 *with 2K buffer* **$2995.00**

Motherboards

ISO-BUS - Jade
Silent, simple, and on sale - a better motherboard
6 Slot (5¼" x 8⅝")
MBS-061B *Bare board* **$19.95**
MBS-061K *Kit* **$39.95**
MBS-061A *A & T* **$49.95**
12 Slot (9¾" x 8⅝")
MBS-121B *Bare board* **$29.95**
MBS-121K *Kit* **$69.95**
MBS-121A *A & T* **$89.95**
18 Slot (14½" x 8⅝")
MBS-181B *Bare board* **$49.95**
MBS-181K *Kit* **$99.95**
MBS-181A *A & T* **$139.95**

Mainframes

MAINFRAME - Cal Comp Sys
12 slot S-100 mainframe with 20 amp power supply
ENC-112105 *Kit* **$309.95**
ENC-112106 *A & T* **$349.95**

DISK MAINFRAME - NNC
Dual 8" drive cutouts with 8 slot motherboard
ENS-112320 *with 30 amp p.s.* **$699.95**

Video Monitors

VIDEO 100 - Leedex
12" B & W video monitor with 12 MHz bandwidth
VDM-801210 .... **$139.95**

VIDEO 100-80 - Leedex
81 x 24 version of Video 100 with metal cabinet
VDM-801230 .... **$189.95**

B & W MONITOR - Sanyo
High quality, high resolution video monitors
VDM-700901 *9" monitor* **$209.95**
VDM-701501 *15" monitor* **$279.95**

13" COLOR MONITOR - Zenith
The hi res color you've been promising yourself
VDC-201301 .... **$449.00**

Call for your free 1980 catalog

Disk Drives

JADE DISK PACKAGE
Double-D controller kit, two 8" double density disk drives, cabinet, power supply, & cables
Special package price **$1295.00**

DUAL 8" DRIVES - Lobo
A pair of double density Shugarts in a cabinet
MSF-12800R *2 single sided* **$995.00**
MSF-125202 *2 double sided* **$1425.00**

HARD DISK - Lobo
10 Megabyte capacity for TRS-80 or Apple systems
MSH-12771C *Apple hard disk* **$4395.00**
MSH-12772C *TRS-80 hard disk* ... **$4395.00**

DISKETTES - Jade
Bargain prices on magnificent magnetic media
5¼" single sided, single density, box of 10
MMD-5110103 *Soft sector* **$27.95**
MMD-5111003 *10 sector* **$27.95**
MMD-5111603 *16 sector* **$27.95**
5¼" double sided, double density, box of 10
MMD-5220103 *Soft sector* **$39.95**
8" single sided, single density, box of 10
MMD-8110103 *Soft sector* **$33.95**
8" single sided, double density, box of 10
MMD-8120103 *Soft sector* **$55.95**
8" double sided, double density, box of 10
MMD-8220103 *Soft sector* **$57.95**

FLOPPY SAVERS - Tri-Star
Protect your valuable software from spindle damage
MMA-205 *5¼" kit* **$13.95**
MMA-208 *8" kit* **$15.95**

Software

CP/M 2.2 - Digital Research
Latest & most powerful release of CP/M
SFC-52506000D *Manual set* **$24.95**
SFC-52506000M *5¼" disk & manual* **$149.95**
SFC-52506000F *8" disk & manual* **$149.95**

MP/M - Digital Research
Multi-user operating system for Z-80 computers
SFC-52507000F *8" disk & manual* **$295.00**

PASCAL/MT - MetaTech
A powerful language for CP/M systems
SFC-73301001F *8" disk & manual* .. **$99.95**

SDOS - SD Systems
DOS, CBASIC-2, Z-80 assembler/editor/linker
SFX-55001000D *Manual set* **$24.95**
SFX-55001002M *5¼" disks & man* **$149.95**
SFX-55001006F *8" disk & manual* **$149.95**

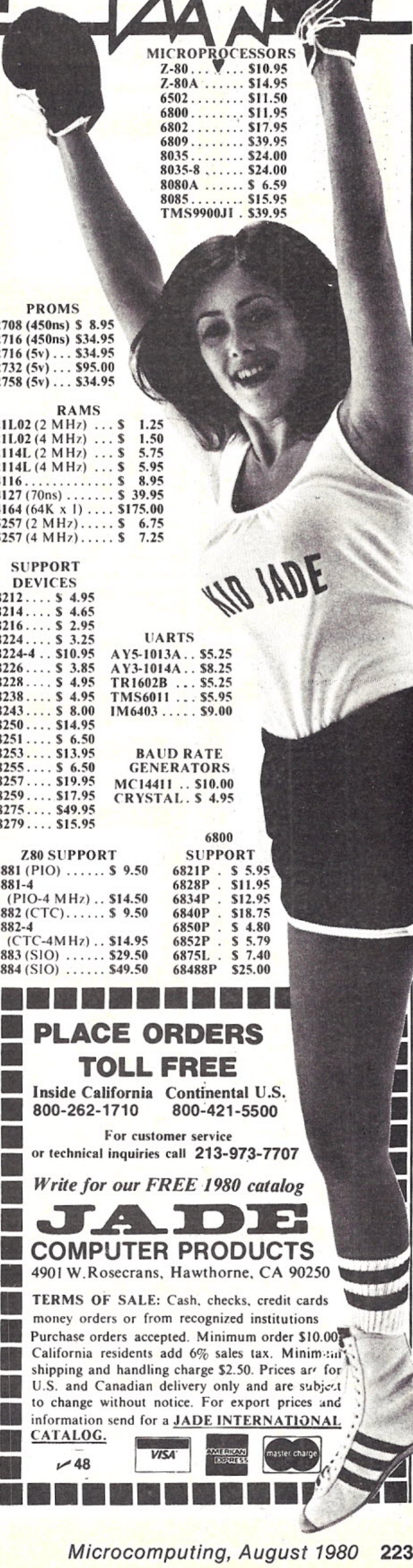
WORDSTAR - MicroPro Intl
The finest word-processing package for CP/M
SFC-13600100F *8" disk & manual* **$395.00**

VISICALC - Personal Software
Visible business/accounting calculator for Apple
SFA-24101005M *5¼" disk & manual* **$145.00**

SINGLE DRIVE COPY - for Apple
Make back-up disks with just a single Disk II
SFA-51150010M *5¼" disk & manual* **$19.95**

SUPER-TEXT - Muse
Professional word-processing package for Apple
SFA-13800085M *5¼" disk & manual* **$99.95**

EPROM ERASER - L.S.Engineering
UV eraser for up to 48 EPROMs
XME-3200 *A & T* **$39.95**



MICROPROCESSORS
Z-80 .... $10.95
Z-80A $14.95
6502 .... $11.50
6800 .... $11.95
6802 .... $17.95
6809 .... $39.95
8035 .... $24.00
8035-8 ...... $24.00
8080A ...... $ 6.59
8085 .... $15.95
TMS9900JI . $39.95

PROMS
2708 (450ns) $ 8.95
2716 (450ns) $34.95
2716 (5v) ... $34.95
2732 (5v) ... $95.00
2758 (5v) ... $34.95

RAMS
21L02 (2 MHz) ... $ 1.25
21L02 (4 MHz) ... $ 1.50
2114L (2 MHz) ... $ 5.75
2114L (4 MHz) ... $ 5.95
4116 .... $ 8.95
4127 (70ns) .... $ 39.95
4164 (64K x 1) $175.00
5257 (2 MHz) ..... $ 6.75
5257 (4 MHz) ..... $ 7.25

SUPPORT DEVICES
8212 .... $ 4.95
8214 .... $ 4.65
8216 .... $ 2.95
8224 .... $ 3.25
8224-4 .. $10.95
8226 .... $ 3.85
8228 .... $ 4.95
8238 .... $ 4.95
8243 .... $ 8.00
8250 .... $14.95
8251 .... $ 6.50
8253 .... $13.95
8255 .... $ 6.50
8257 .... $19.95
8259 .... $17.95
8275 .... $49.95
8279 .... $15.95

UARTS
AY5-1013A .. $5.25
AY3-1014A .. $8.25
TR1602B ... $5.25
TMS6011 ... $5.95
IM6403 ..... $9.00

BAUD RATE GENERATORS
MC14411 .. $10.00
CRYSTAL . $ 4.95

Z80 SUPPORT
3881 (PIO) ...... $ 9.50
3881-4 (PIO-4 MHz) .. $14.50
3882 (CTC) ...... $ 9.50
3882-4 (CTC-4MHz) .. $14.95
3883 (SIO) $29.50
3884 (SIO) $49.50

6800 SUPPORT
6821P . $ 5.95
6828P . $11.95
6834P . $12.95
6840P . $18.75
6850P . $ 4.80
6852P . $ 5.79
6875L . $ 7.40
68488P $25.00

PLACE ORDERS TOLL FREE

Inside California 800-262-1710
Continental U.S. 800-421-5500

For customer service or technical inquiries call **213-973-7707**

Write for our FREE 1980 catalog

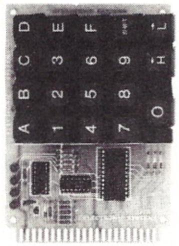
JADE COMPUTER PRODUCTS
4901 W.Rosecrans, Hawthorne, CA 90250

TERMS OF SALE: Cash, checks, credit cards money orders or from recognized institutions Purchase orders accepted. Minimum order $10.00. California residents add 6% sales tax. Minimum shipping and handling charge $2.50. Prices are for U.S. and Canadian delivery only and are subject to change without notice. For export prices and information send for a **JADE INTERNATIONAL CATALOG.**

48

VISA · AMERICAN EXPRESS · master charge

HEX ENCODED KEYBOARD



Four onboard LEDs indicate the HEX code generated for each key depression. The board requires a single +5 volt supply. Board only $15.00 Part No. HEX-3, with parts $49.95 Part No. HEX- 3A. 44 pin edge connector $4.00 Part No. 44P.

ASCII TO CORRESPONDENCE CODE CONVERTER

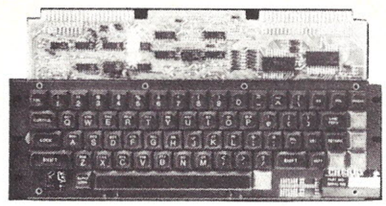
This bidirectional board is a direct replacement for the board inside the Trendata 1000 terminal. The on board connector provides RS-232 serial in and out. Sold only as an assembled and tested unit for $249.95. Part No. TA 1000C

ASCII KEYBOARD

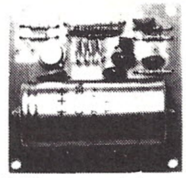
53 Keys popular ASR-33 format • Rugged G-10 P.C. Board • Tri-mode MOS encoding • Two-Key Rollover • MOS/DTL/TTL Compatible • Upper Case lockout • Data and Strobe inversion option • Three User Definable Keys • Low contact bounce • Selectable Parity • Custom Keycaps • George Risk Model 753. Requires +5, -12 volts. $59.95 Kit.

ASCII KEYBOARD

TTL & DTL compatible • Full 67 key array • Full 128 character ASCII output • Positive logic with outputs resting low • Data Strobe • Five user-definable spare keys • Standard 22 pin dual card edge connector • Requires +5VDC, 325 mA. Assembled & Tested. Cherry Pro Part No. P70-05AB. $119.95.

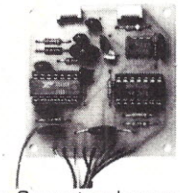


T.V. INTERFACE



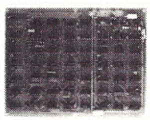
● Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple ● Power required is 12 volts AC C.T., or +5 volts DC ● Board only $7.60 part No. 107, with parts $13.50 Part No. 107A

TAPE INTERFACE



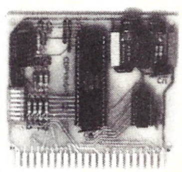
● Converts a low cost tape recorder to a digital recorder ● Works up to 1200 baud ● Digital in and out are TTL-serial ● Output of board connects to mic. in of recorder ● Earphone of recorder connects to input on board ● No coils ● Requires +5 volts, low power drain ● Board only $7.60 Part No. 111, with parts $29.95Part No. 111A

T.V. TYPEWRITER



● Stand alone TVT ● 32 char/line, 16 lines, modifications for 64 char/line included ● Parallel ASCII (TTL) input ● Video output ● 1K on board memory ● Output for computer controlled curser ● Auto scroll ● Non-destructive curser ● Curser inputs: up, down, left, right, home, EOL, EOS ● Scroll up, down ● Requires +5 volts at 1.5 amps, and -12 volts at 30 mA ● All 7400, TTL chips ● Char. gen. 2513 ● Upper case only ● Board only $39.00 Part No. 106, with parts $145.00 Part No. 106A

UART & BAUD RATE GENERATOR

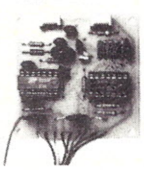


● Converts serial to parallel and parallel to serial ● Low cost on board baud rate generator ● Baud rates: 110, 150, 300, 600, 1200, and 2400 ● Low power drain +5 volts and -12 volts required ● TTL compatible ● All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. ● All connections go to a 44 pin gold plated edge connector ● Board only $12.00 Part No. 101, with parts $35.00 Part No. 101A, 44 pin edge connector $4.00 Part No. 44P

SOROC IQ 120

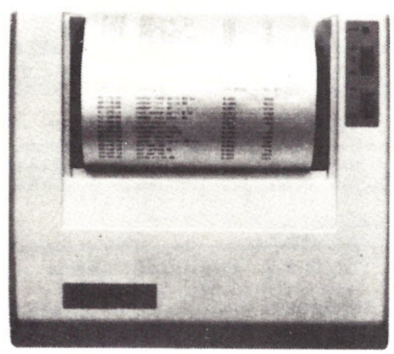
Upper/lower case display • Numeric keypad & cursor keys • Protected fields, 1/2 intensity display • RS 232 interface & aux. port. IQ120—$799.95 • IQ140 Detachable keyboard—$1199.95

MODEM



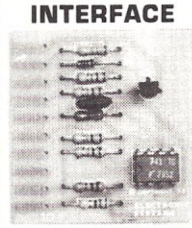
● Type 103 ● Full or half duplex ● Works up to 300 baud ● Originate or Answer ● Serial TTL input and output ● connect 8 Ω speaker and crystal mic. directly to board ● Requires +5 volts ● Board only $7.60 Part No. 109, with parts $29.95 Part No. 109A.

COMPRINT PRINTER



Printing Characteristics: 225 characters/second (170 lines/minute) throughput ● 9 horizontal x 12 vertical matrix ● 96 ASCII character set with upper and true lower case ● 80 characters/line ● 5.8 lines/inch
Buffer Memory: standard 256 bytes; ● optional; 2,048 bytes (buffer memory option designated as Model 912-2K), add $149.95.
Paper Requirements: electrosensitive type (aluminum coated) ● 8-1/2 inch width ● 3.7 inch max. (300 ft.) roll diameter.
Model 912-S Interfacing: serial interface RS232 and 20 mA current loop ● BAUD rates 110, 150, 300, 600, 1200, 2400 and 4800 are strap selectable.
Model 912-P Interfacing: parallel interface, IEEE-488 and 8 bit parallel (strobe/ acknowledge). Model 912-S, Part No. CPIA, 32118, $579.95. Model 912-P, Part No. CPIA, 32117, $559.95.

RS-32/TTL INTERFACE



● Converts TTL to RS-232, and converts RS-232 to TTL ● Two separate circuits ● Requires -12 and +12 volts ● All connections go to a 10 pin edge connector, kit $9.95 Part No. 232A 10 Pin edge connector $3.00 part No. 10P.

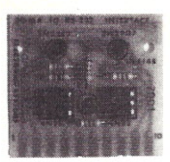
COMPUCOLOR II

With reg. keyboard
MOD3 8K $1595.95
MOD4 16K $1695.95
MOD 5 32K $1995.95
Now includes $250 more, worth of software and accessories with 101 key option add $134.95 with 117 key option add $179.95

44 BUS MOTHER BOARD

Has provisions for ten 44 pin (.156) connectors, spaced 3/4 of an inch apart. Pin 20 is connected to X, and 22 is connected to Z for power and ground All the other pins are connected in parallel. This board also has provisions for bypass capacitors. Board cost $15.00 Part No. 102. Connectors $3.00 each Part No. 44WP.

RS-232/20mA INTERFACE



This board has two passive, opto-isolated circuits. One converts RS-232 to 20mA, the other converts 20mA to RS-232. All connections go to a 10 pin edge connector. Requires +12 and -12 volts. Board only $9.95, part no. 7901, with parts $14.95 Part No. 7901A.

DC POWER SUPPLY

• Board supplies a regulated +5 volts at 3 amps., +12, -12, and -5 volts at 1 amp. • Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps. • Board only $12.50 Part No. 6085, with parts excluding transformers $42.50 Part No. 6085A

To Order: Mention part no. description, and price. In USA shipping paid by us for orders accompanied by check or money order. We accept C.O.D. orders in the U.S. only, or a VISA or Master Charge no., expiration date, signature, phone no., shipping charges will be added. CA residents add 6.5% for tax. Outside USA add 10% for air mail postage and handling. Payment must be in U.S. dollars. Dealer inquiries invited. 24 hour order line (408) 448-0800 ✓47

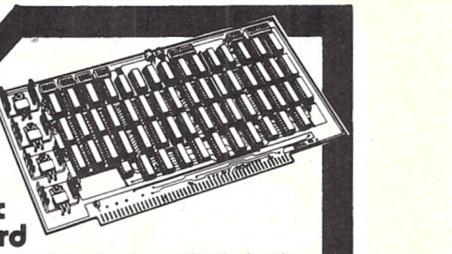
Send for FREE Catalog . . . a big self-addressed envelope with 41¢ postage gets it fastest!

ELECTRONIC SYSTEMS Dept. KB P. O. Box 21638, San Jose, CA USA 95151

HOBBYWORLD® ELECTRONICS, INC.

✓10

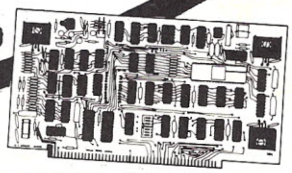
Call Toll-Free: USA (800) 423-5387
In California: (800) 382-3651
Local & Outside USA: (213) 886-9200



CCS 16K Static Ram Board

True static RAM board designed specifically for the S-100 bus. Requires only +5VDC. Features true static operation, and all bus signals labeled on-board! Uses 2114 low power static RAS. S-100 compatible, fully buffered. Silk screened PC board, solder masked on both sides.

Cat No.	Description	Wt	Price
1601A	16K RAM 450ns Kit	1 lb	$253.50
1601B	16K RAM 200ns Kit	1 lb	$333.50
1601C	16K RAM 300ns Kit	1 lb	$293.50
1602A	16K RAM 450ns A & T	1 lb	$285.75
1602B	16K RAM 200ns A & T	1 lb	$371.50
1602C	16K RAM 300ns A & T	1 lb	$328.50
1603	16K RAM Bareboard only	6 oz.	$ 29.95



Microsoft TRS-80 FORTRAN +

Fortran is now available for TRS-80! Because Fortran Plus is a compiler and will support any I/O, most level II BASIC limitations are eliminated. Create your own subroutines, assembler source files, data files and FORTRAN files. Fully compatible with TRS DOS, the FORTRAN compiler can operate 1200 lines per minute on a single pass. Requires a minimum 32K TRS-80 disk system. Compiler also generates a fully symbolic listing of the machine language that is generated. Also included is a Macro assembler that accepts Z80 opcodes and supports complete Intel standard macro facility. Linking loader provides a variety of capabilities, which are executed by means of easy command lines and switches. Any number of programs may be loaded with just one command.

A comprehensive library of subroutines is supplied. Most of the popular commands as well as a few special ones are included. Only the library routines required to run a particular FORTRAN program need be loaded before execution. Users may write non-standard I/O drivers for each Logic Unit Number, making the task of interfacing non-standard devices to TRS-80 FORTRAN programs straightforward.

Cat No. 1341 **$185.**

SSM SB 1 MUSIC SYNTHESIZER BOARD

Wave form synthesizer, designed to interface with the S-100 bus. Can be programmed to play a monoghonic solo in an instrumental voice that is controlled by the software. Software includes a timer for definition of tempo and time signatures. Music can be heard immediately after the notation has been typed in. Multiple cards can be used to produce chords and harmonies. Weight 1 lb.

Cat No.	Description	Price
1408	Kit with Software	$199
1409	SB 1 Assembled & Tested	$279
1410	SB 1 bare board	$ 45

CCS 32K Static RAM Board $710

Uses 2114, 250ns fully Static Rams, bank selectable in 8K blocks. Enable/Disable on Power up or Reset. Compatible with North Star, Alpha Micro, Cromeco, etc. Also front panel compatible, addressable in 8K blocks. Selectable Wait state. Wt 1 lb. Assembled & tested.
Cat No. 2644

MICROPOLIS® DISK DRIVES

Imagine a 5¼" floppy disk system with all the storage capacity of an 8" floppy system, and more!

Micropolis can give you more storage because they pack more data onto every disk. Ordinary 5¼" floppies provide just 35 tracks per drive and store 70 to 130 K bytes of data. Instead, Micropolis uses 77 tracks-each with 16 sectors of 156 bytes to yield an incredible storage capacity of 315k bytes per drive.

And that's not all! Reliability doesn't just happen by accident. At Micropolis reliability is engineered into each step of manufacturing. For example, most 5¼" floppy disks cut costs by using a plastic cam or cam follower to position the read/write head. Micropolis chose to use the strength and durability of an all steel cam and cam follower. Sure, it costs more, but it gives you more accurate tracking over a significantly greater lifespan which adds up to a lower cost per byte with use. Software from Micropolis includes a comprehensive DOS, (disk operating system), and disk Extended Basic designed for 8080/Z80 microcomputer systems. The DOS is complete with an assembler, editor, file management functions and disk utilities. Micropolis BASIC is a complete, powerful programming tool for developing, testing, executing and maintaining basic programs.

The model 1043 MOD II is a single floppy disk with 315k bytes storage and includes the S-100 disk Controller board. If you need more storage, or simply want to save even more money, then order the model 1053 MOD II dual disk system with 630k bytes storage capacity and S-100 Controller board. Micropolis DOS and disk extended BASIC are standard with both units.

Cat No.	Description	Weight	Price
2563	Model 1043 mod II single drive	9 lbs	$1145.00
2564	Model 1053 mod II dual drive	18 lbs	$1895.00

THE PIE 2.0 $79.95 (Programma Improved Editor)

Don't be mislead by the low price of this outstanding wordprocessing package. PIE 2.0 is a powerful text editor and print format processor that has all the bells and whistles expected of wordprocessing software costing three times as much. Some features include:

1) Character/line insert and delete
2) Complete Cursor mobility
3) String search forward and backward
4) Single, conditional or global search and replace
5) Move and/or copy blocks of text
6) Page scrolling
7) Tabs, margins, paragraphing, etc.

Research conducted by IBM Corp. revealed that the time required to create, edit and complete a one page document was decreased by as much as 60% when comparing the performance of a Wordprocessing system to an ordinary typewriter. Finding ways to remain competitive these days is a challenge for the business executive. Todays office can substantially improve their daily productivity level with PIE 2.0 Wordprocessing software and an Apple II computer with 32k RAM memory.

As a businessman you want every dollar you spend to count, so wordprocessing makes sense, and PIE 2.0 Wordprocessing software gives you more for your hard earned dollar. PIE 2.0 Wordprocessing software comes complete with program diskette and detailed documentation in a handsome, simulated leather binder.
Cat No. 2562 Apple II, 32K

16K MEMORY ADD-ON KIT $55.00

Everything you need to upgrade your TRS-80, Apple or Exidy! An additional 16K includes illustrated instructions, RAMS and preprogrammed jumpers. No special tools required. Wt. 4 oz.

CAT NO	DESCRIPTION
1156	TRS-80 Keyboard Unit
1156-A	TRS-80 Exp. Interface (prior to 4/1/79)
1156-B	TRS-80 Exp. Interface (after 4/1/79)
1156-C	for Apple II
1156-D	for Exidy

VERBATIM 5¼" DISKETTES 10 per box

CAT NO.	TYPE	DESCRIPTION	PRICE
1147	525-01	Soft sector, TRS-80, etc.	$33.00
1148	525-10	10 hole, hard, Apple, North Star	$33.00
1149	525-16	16 hole, hard, Micropolis	$33.00
2330	577-01	Soft sector, certified	$49.95
2331	577-10	10 hole, hard, certified	$49.95
2332	577-16	16 hole, hard, certified	$49.95

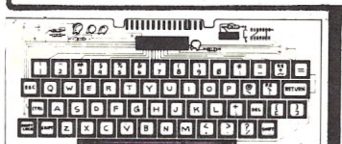
DISKETTE DRIVE HEAD CLEANING KIT

Diskette drive heads need periodic maintenance to assure efficient and error-free operation. Unlike other peripheral devices, the read/write head(s) on diskette drives are extremely difficult to clean without partially disassembling the drive. The diskette drive head cleaning kit allows the user to clean the heads in just minutes, without disassembling the drive. Available in 5½" or 8", both single and double sided. Kit contains 2 cleaning diskettes, 4 oz. bottle of CS-85 cleaning solution and an easy pour dispenser.
Weight 12 oz.

CAT NO.	DESCRIPTION	PRICE
2499	8" Disk Drive Cleaning Kit	$30.75
2534	5¼" Disk Drive Cleaning Kit	$30.75

56 Key ASCII Keyboard

Professional quality at an affordable price. Interfaces with all computers requiring 7-bit standard ASCII code. Packed with features such as tri-mode MOS encoding, positive or negative logic output, pulsed and level strobe, and two key rollover! Compatible with CMOS, TTL, and DTL circuitry. On-board electronic shift-lock. Easy to assemble, with full documentation.



Cat No.	Description	Wt.	Price
1143	Keyboard kit	2 lb	$63.00
1144	Keyboard a & t	2 lb	$74.50
1548	All metal, pre-cut enclosure	3 lb	$30.75
1144A	Edge connector for keyboard	8 oz	$ 3.50

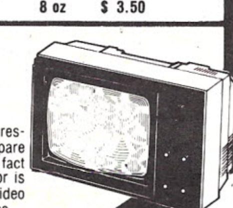
HOW TO ORDER

Pay by check, Mastercharge, Visa, or C.O.D. Charge card orders please include expiration date. Payment in U.S. dollars only. Order by phone, mail or at our retail store. MINIMUM ORDER $10.00. Please include phone number and magazine issue you are ordering from. Prices valid thru last day of cover date. SHIPPING: USA: Add $2.00 for first 2 lbs., 35¢ each addt'l lb. for ground. For AIR add $3.00 first 2 lbs., 75¢ each addt'l lb. FOREIGN: surface: $3.00 first 2 lbs., 60¢ each addt'l lb. AIR: $11.00 first 2 lbs., $5.00 each addt'l lb. CODs: add $1.25 addt'l. Not responsible for typographical errors. Some items subject to prior sale or quantity limitations. 120 day guaranteed satisfaction. Exception: partially assembled kits.

SEND FOR FREE FLYER, FEATURING:

Page after page of exciting products. Computerized toys and games, personal computers, disk drives, integrated circuits, semi conductors. Add new dimension to your Apple, Atari, TRS-80, etc. with our special application boards and comprehensive software library. Hundreds of products available at terrific Hobbyworld prices. Circle our reader service number or write/phone for your free illustrated flyer today.

Leedex VIDEO 100 12" MONITOR



- *Compatible with TRS-80 (no interface required)*
- *Compatible with many home computers!*

Now from LEEDEX . . . One of the most popular low cost, yet high resolution (650 line) monitors currently available. These units compare favorably with monitors costing twice as much. Because of the fact that standard composite video input is utilized no RF modulator is needed. An extremely sharp and stable picture is achieved. The video bandwidth is 12 mhz +/− 3db with a 75 ohm input impedance.

Cat No.	Description	Wt.	Price
1204	Video 100 Monitor	18 lb.	$149.00
1937	TRS-80/Leedex cable kit	6 oz.	$3.00

Dept. K8

19511 Business Center Drive
Northridge, Calif. 91324

ADVANCED COMPUTER PRODUCTS

✓51

7400

SN7400N	.19	SN74123N	.59
SN7401N	.22	SN74125N	.39
SN7402N	.22	SN74126N	.44
SN7403N	.22	SN74128N	.59
SN7404N	.22	SN74132N	.69
SN7405N	.23	SN74136N	.95
SN7406N	.23	SN74139N	.95
SN7407N	.23	SN74141N	.69
SN7408N	.26	SN74142N	2.95
SN7409N	.23	SN74143N	2.95
SN7410N	.22	SN74144N	2.95
SN7411N	.29	SN74145N	.62
SN7412N	.29	SN74147N	1.95
SN7413N	.39	SN74148N	1.20
SN7414N	.59	SN74150N	.99
SN7416N	.29	SN74151N	.67
SN7417N	.29	SN74152N	.67
SN7420N	.22	SN74153N	.67
SN7421N	.35	SN74154N	1.19
SN7422N	.29	SN74155N	.82
SN7423N	.29	SN74156N	.89
SN7425N	.29	SN74157N	.69
SN7426N	.29	SN74158N	1.65
SN7427N	.29	SN74160N	.95
SN7429N	.45	SN74161N	.95
SN7430N	.23	SN74162N	.89
SN7432N	.29	SN74163N	.87
SN7437N	.29	SN74164N	.97
SN7438N	.29	SN74165N	.97
SN7439N	.29	SN74166N	1.20
SN7440N	.24	SN74167N	1.95
SN7441N	.79	SN74170N	1.69
SN7442N	.57	SN74172N	5.95
SN7443N	.79	SN74173N	.79
SN7444N	.79	SN74174N	.89
SN7445N	.79	SN74175N	.89
SN7446N	.79	SN74176N	.85
SN7447N	.59	SN74177N	.85
SN7448N	.79	SN74179N	1.80
SN7450N	.23	SN74180N	.75
SN7451N	.23	SN74181N	1.75
SN7453N	.23	SN74182N	.75
SN7454N	.23	SN74184N	1.95
SN7459N	.29	SN74185N	1.95
SN7460N	.23	SN74186N	9.95
SN7470N	.39	SN74188N	3.90
SN7472N	.34	SN74190N	1.15
SN7473N	.38	SN74191N	1.15
SN7474N	.36	SN74192N	.85
SN7475N	.38	SN74193N	.85
SN7476N	.36	SN74194N	.85
SN7479N	4.60	SN74195N	.85
SN7480N	.59	SN74196N	.85
SN7481N	1.10	SN74197N	.85
SN7482N	1.10	SN74198N	1.39
SN7483N	.55	SN74199N	1.39
SN7485N	.65	SN74221N	1.39
SN7486N	.39	SN74251N	.95
SN7489N	1.75	SN74273N	1.05
SN7490N	.39	SN74279N	.89
SN7491N	.65	SN74283N	2.15
SN7492N	.52	SN74284N	3.90
SN7493N	.49	SN74285N	3.90
SN7494N	.72	SN74290N	1.25
SN7495N	.65	SN74298N	.95
SN7496N	.72	SN74365N	.68
SN7497N	3.10	SN74366N	.68
SN74100N	.99	SN74367N	.79
SN74107N	.32	SN74368N	.79
SN74109N	.53	SN74390N	1.90
SN74116N	1.95	SN74393N	1.90
SN74121N	.29	SN74490N	1.90
SN74122N	.39		

CMOS

CD4000	.29	CD4093	.99
CD4001	.29	CD4094	2.95
CD4002	.29	CD4098	2.49
CD4006	1.39	CD4099	2.25
CD4007	.29	MC14408	12.95
CD4008	1.39	MC14409	12.95
CD4009	.59	MC14410	12.95
CD4010	.59	MC14412	12.95
CD4011	.29	MC14415	8.95
CD4012	.29	MC14419	4.95
CD4013	.49	CD4501	.39
CD4014	1.39	CD4502	1.65
CD4015	1.39	CD4503	.69
CD4016	.59	CD4505	8.95
CD4017	1.19	CD4506	.75
CD4018	1.19	CD4507	.95
CD4019	.49	CD4508	3.95
CD4020	1.19	CD4510	1.39
CD4021	1.49	CD4511	1.39
CD4022	1.29	CD4512	1.39
CD4023	.38	CD4515	3.95
CD4024	.79	CD4516	1.69
CD4025	.38	CD4518	1.39
CD4027	.79	CD4520	1.39
CD4028	.99	CD4555	4.95
CD4029	1.29	CD4556	.99
CD4030	.69	CD4566	2.25
CD4031	3.25	74C00	.39
CD4032	2.15	74C02	.39
CD4034	3.25	74C04	.39
CD4035	1.19	74C08	.49
CD4037	1.95	74C10	.49
CD4040	1.29	74C14	1.65
CD4041	1.25	74C20	.39
CD4042	.99	74C30	.39
CD4043	.99	74C32	.99
CD4044	.99	74C42	1.85
CD4046	2.25	74C48	2.39
CD4047	1.25	74C73	.99
CD4048	.69	74C74	.99
CD4049	.69	74C85	2.49
CD4050	.69	74C89	4.95
CD4051	1.10	74C90	1.85
CD4052	1.10	74C93	1.85
CD4053	1.10	74C95	1.85
CD4055	3.95	74C107	1.19
CD4056	2.95	74C151	2.49
CD4059	9.95	74C154	3.50
CD4060	1.39	74C157	2.10
CD4066	.89	74C160	2.39
CD4069	.35	74C161	2.30
CD4070	.69	74C163	2.39
CD4071	.35	74C164	2.39
CD4072	.35	74C173	2.59
CD4073	.35	74C174	2.75
CD4075	.35	74C175	2.75
CD4076	1.29	74C192	2.39
CD4077	.35	74C193	2.39
CD4078	.35	74C195	2.39
CD4081	.35	74C922	7.95
CD4082	.35	74C923	6.95
CD4085	1.95	MM80C95	1.50
CD4089	2.95	MM80C97	1.25

74LS00

74LS00N	.35	74LS164N	1.19
74LS01N	.28	74LS165N	.89
74LS02N	.28	74LS166N	2.48
74LS03N	.28	74LS168N	1.89
74LS04N	.39	74LS169N	1.89
74LS05N	.28	74LS170N	1.99
74LS08N	.39	74LS173N	.89
74LS09N	.39	74LS174N	.99
74LS10N	.28	74LS175N	.89
74LS11N	.39	74LS181N	2.20
74LS12N	.39	74LS190N	1.15
74LS13N	.47	74LS191N	1.15
74LS14N	1.25	74LS192N	.98
74LS15N	.39	74LS193N	.98
74LS20N	.26	74LS194N	1.15
74LS21N	.38	74LS195N	.95
74LS22N	.38	74LS196N	.89
74LS26N	.39	74LS197N	.89
74LS27N	.39	74LS221N	1.49
74LS28N	.39	74LS240N	2.99
74LS30N	.26	74LS241N	2.49
74LS32N	.39	74LS242N	2.29
74LS37N	.79	74LS243N	2.29
74LS38N	.39	74LS244N	2.95
74LS40N	.26	74LS245N	8.95
74LS42N	.79	74LS247N	1.10
74LS47N	.79	74LS248N	1.10
74LS48N	.79	74LS249N	1.69
74LS51N	.26	74LS251N	1.79
74LS54N	.35	74LS253N	.98
74LS55N	.35	74LS257N	.98
74LS73N	.45	74LS258N	.98
74LS74N	.59	74LS259N	2.95
74LS75N	.68	74LS260N	.69
74LS76N	.45	74LS261N	2.49
74LS78N	.65	74LS266N	.59
74LS83AN	.99	74LS273N	1.75
74LS85N	1.19	74LS275N	4.40
74LS86N	.45	74LS279N	.59
74LS90N	.75	74LS283N	1.10
74LS92N	.75	74LS290N	1.29
74LS93N	.75	74LS293N	1.95
74LS95N	.88	74LS295N	1.10
74LS96N	.98	74LS298N	1.29
74LS107N	.45	74LS324N	1.75
74LS109N	.45	74LS347N	1.95
74LS112N	.49	74LS348N	1.95
74LS113N	.49	74LS352N	1.65
74LS114N	.55	74LS353N	1.65
74LS122N	.55	74LS363N	1.49
74LS123N	1.19	74LS365N	.99
74LS124N	1.35	74LS366N	.99
74LS125N	.89	74LS367N	.99
74LS126N	.89	74LS368N	.99
74LS132N	.79	74LS373N	2.75
74LS136N	.59	74LS374N	2.75
74LS138N	.89	74LS375N	.69
74LS139N	.89	74LS377N	1.95
74LS145N	1.25	74LS385N	1.95
74LS148N	1.49	74LS386N	.65
74LS151N	.79	74LS390N	1.95
74LS153N	.79	74LS393N	1.95
74LS154N	2.49	74LS395N	1.70
74LS155N	1.19	74LS399N	2.95
74LS156N	.99	74LS424N	2.95
74LS157N	.99	74LS668N	1.75
74LS158N	.75	74LS670N	2.29
74LS160N	.98	81LS95N	1.99
74LS161N	1.15	81LS96N	1.99
74LS162N	.98	81LS97N	1.99
74LS163N	.98	81LS98N	1.99

LINEAR

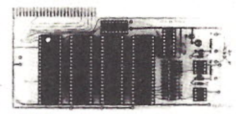
78H05	5.95	LM1414N	1.90
78M06	1.49	LM1458CN/N	.49
78M.G.	1.49	MC1488N	1.49
LM105H	.99	MC1489N	1.49
LM108AH	2.95	LM1496N	.89
LM300H	.79	LM1556N	1.50
LM301CN/H	.35	LM1800N	.79
LM304H	.98	LM1820N	.95
LM305H	.89	LM1850N	.95
LM306H	3.25	LM1889N	3.95
LM307CN/H	.29	LM2111N	1.75
LM308CN/H	.98	LM2900N	.99
LM309K	1.49	LM2901N	2.50
LM310CN	1.25	LM2917N	2.95
LM311D/CN/H	.98	CA3013T	2.29
LM312H	1.75	CA3018T	1.99
LM317T	2.75	CA3021T	3.49
LM318CN/H	1.49	CA3023T	2.99
LM319N/H	1.25	CA3035T	2.75
LM320K-XX*	1.49	CA3039T	1.49
LM320T-XX*	1.25	CA3046T	1.29
LM320H-XX*	1.25	LM3053N	1.49
LM323K	4.95	CA3059N	3.25
LM324N	1.25	CA3060N	3.25
LM339N	.95	CA3062N	4.95
LM340K-XX*	1.49	LM3065N	1.49
LM340T-XX*	1.25	CA3080N	1.29
LM340H-XX*	1.25	CA3081N	1.69
LM344H	1.95	CA3082N	1.69
LM348N	1.85	CA3083N	1.99
LM358CN	.98	CA3086N	1.29
LM360N	1.49	CA3089N	2.75
LM372N	1.95	CA3096N	2.49
LM376N	3.75	CA3097N	1.99
LM377N	3.75	CA3130T	2.49
LM380CN/N	1.25	CA3140T	2.49
LM381N	1.79	CA3146N	2.49
LM383T	1.95	CA3160T	1.49
LM386N	1.49	CA3190N	1.95
LM387N	1.49	CA3401N	.69
LM390N	1.95	MC3423N	1.49
NE531V/T	3.75	MC3460N	3.95
NE555V	.39	SG3524N	3.95
NE556N	.98	CA3600N	3.50
NE561T	19.95	LM3900N	.59
NE562B	7.95	LM3905N	1.49
NE565N/H	1.25	LM3909N	.98
NE566H/V	1.75	RC4131N	2.95
NE567V/H	1.50	RC4136N	1.10
NE592N	2.75	RC4151N	4.50
LM702H	2.99	RC4194	4.95
LM709N/H	.29	RC4195	4.40
LM710N/H	.98	ULN2001	1.25
LM711N/H	.39	ULN2003	1.50
LM715N	1.95	SN75450N	.59
LM723N/H	.75	SN75451N	.49
LM733N/H	.98	SN75452N	.49
LM739N	1.15	SN75453N	.49
LM741CN/H	.33	SN75454N	.49
LM741CN-14	.19	SN75491N	.89
LM747N/H	.79	SN75492N	.89
LM748N/H	.39	SN75493N	.89
LM760CN	2.05	SN75494N	.89
LM1310N	1.90		

BECKMAN
Digital Multimeters
MODELS TECH 300 AND TECH 310

AS LOW AS **$100.00**

Choice of Models – The TECH 310 has all above features, 7 functions, 29 ranges plus 0.25% Vdc accuracy.
The TECH 300 has a 0.5% Vdc accuracy and all the above features, but without Insta-Ohms™ continuity function or the 10 amp current ranges.

TECH 300 Digital Multimeter .... $100.
TECH 310 Digital Multimeter .... $130.
VC-201 Vinyl Carrying Case............ $10.
DC-202 Deluxe Carrying Case .... $24.
HV-211 High Voltage Probe............ $35.
RP-221 RP Probe............ $35.
Ct-231 AC Current Clamp............ $42.
DL-241 Deluxe Test Lead Kit............ $10.
TL-242 Spare Test Leads .... $6. (6 pcs.)



ACP APPLE MUSIC MACHINE
WITH 9 VOICES!

• NEW! Uses latest State of the Art LSI Technology • Requires only one slot for 9 voices • Uses three AY3-8910's to produce nine voices (Other competitive models have only 3 voices) • Includes software • Simulates three ALF Boards • Plays music generated by the ALF Board • APPLE™ II compatible.

3 Times More Powerful Than ALF!

KIT $199.95
ASSEMBLED AND TESTED $269.95

THE BONE FONE

- SKIERS
- JOGGERS
- SKATERS
- CYCLERS

You must hear it and feel it to believe it!

AM/FM stereo surrounds and fills your body with sound. No earplugs. You wear it!

$65.95

FLOPPY DISK DRIVES

MPI B51-5¼", 40 tracks 279.00
Shugart SA400-5¼" 35 tracks 295.00
Shugart 800/801R 8" 475.00
Siemens Shugart Compatible Model FDD-120-8D 429.00
PERSCI Model 277 Dual 1195.00
WANGO/SIEMENS 5¼" Drive 290.00
MPI B52 5¼" Dual 395.00
WANGO/SIEMENS 282 Dual 5¼" 395.00
WANGO/SIEMENS 82 .... 290.00

MONITORS

Sanyo 9" $169.95
Sanyo 15" 279.00
Leedex 12" 139.95
Motorola 12", High Resolution, 22 MHz, OEM Model #M3000-340 .... 219.00
Zenith 13" Color Monitor... 499.00
MGA 13" Color TV 349.00
VAMP 19" Color Monitor... 575.00
VAMP 15" Color Monitor... 449.00

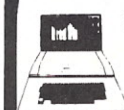
CSC CONTINENTAL SPECIALTIES

Model 3001 Digital Capacitance Meter.... 275.00
Model 333 Tri-Mode Comparator 295.00
Model LM-3 40-channel Logic Monitor.... 585.00
Model LM-1 Logic Monitor............ 60.00
Model LM-2 Logic Monitor............ 147.00
Model 2001 Sweepable Function Generator... 186.00
Model 5001 Universal Counter-Timer 360.00
Model 6001 650 MHz Frequency Counter... 385.00
MAX-100 100 MHz Portable Frequency Counter 149.00
PS-500 500 MHz Decade Prescaler........ 70.00
MAX-50 50 MHz Handheld Frequency Counter 77.00
MAX-550 550 MHz Handheld Frequency Counter 165.00
Model 4001 Pulse Generator............ 835.00
Model DP-1 Digital Pulser 83.00
Logic Probes............
Model LP-1 Digital Logic Probe............ 50.00
Model LP-2 Economy Logic Probe............ 28.00
Model LP-3 High Speed Logic Probe........ 77.00
Model LPK-1 Logic Probe Kit............ 21.95
Logic Probe Accessories 21
Model LTC-1, LTC-2 Logical Analysis Kits... 220/250

Apple II. 16K or Apple II. Plus $990

16K Apple Upgrade Kit $62.95

Hi-Speed Serial I/O	$189.00	DS65 Digi-Sector	$349.00	Integer ROM Card	$189.00
Centronics Printer I/O	199.95	Apple Graphics Tablet	725.00	Proto Card	21.95
Applesoft II Firmware	189.00	D.C. Hayes Modem II	349.95	M & R Modulator	29.95
Apple Clock	269.00	Disk II w/Controller	575.00	Sanyo Cassette	54.95
Introl X-10 System	269.95	Disk II	475.00	16K Upgrade Kit	62.95
Introl X-10 Controller	179.95	Pascal Lang. System	450.00	Desktop Plan	98.00
Alf Music Synthesizer	259.95	Parallel Printer Card	165.00	8" Floppy Controller	350.00
16 Ch. Analog Input	249.95	Communications Card	189.00	Heuristics Speechlab	179.00
2 Ch. Analog Output	164.95	Business Software Pkg.	625.00	Romplus +	169.00
13-Key Keypad	119.95	Corvus 10 Megabyte Drive	4625.00	Supertalker	279.00
Visi-Calc	125.00			Cashier	250.00



ATARI® 800 & 400
Personal Computer System

ATARI 800 $750.00
ATARI 400 $449.00

ATARI 800 Includes: Computer Console, BASIC Lang. Cartridge, Education System Master Cartridge, BASIC Language Programming Manual, 800 Operator's Manual w/Notebook, Atari 410 Program Recorder, 8K RAM Module, Power Supply, TV Switch Box.

Peripherals		**Education System Cassette Programs**		Spelling	35.00
Disc Drive	599.00	U.S. History	35.00	Basic Electricity	35.00
Printer	499.00	U.S. Government	35.00	Basic Algebra	35.00
Program Recorder	69.00	Supervisory Skills	35.00	**Basic Game and Program Cassettes**	
Software – ROM Cartridges		World History (Western)	35.00	Guide to BASIC Programming	18.00
Education System Master Cartridge	28.00	Basic Sociology	35.00	BASIC Game Programs	19.00
Atari BASIC	55.00	Counseling Procedures	35.00	**Add-On Memory**	
Assembler Debug	55.00	Principles of Accounting	35.00	8K RAM Memory Module	110.00
Basketball	42.00	Physics	35.00	16K RAM Memory Module	225.00
Life	42.00	Great Classics (English)	35.00	**Diskettes**	
Super Breakout™	42.00	Business Communications	35.00	Blank Diskettes	
Music Composer	55.00	Basic Psychology	35.00	Disk File Manager	5.00
Super Bug™	42.00	Effective Writing	35.00	**Accessory Controllers**	
Computer Chess	55.00	Auto Mechanics	35.00	Driving Controller Pair	18.00
Home Finance	55.00	Principles of Economics	35.00	Paddle Controller Pair	18.00
				Joystick Controller Pair	18.00

TEXAS INSTRUMENTS
INCORPORATED

99/4 PERSONAL COMPUTER

Superior Color, Music, Sound and Graphics – and a Powerful Extended Basic – All Built In.

TI 99/4 Console only available for $698.90.

$1099.00

commodore PET

AS LOW AS **$775.00**

DISCOUNT PRICES

2001-8K	775.00	2023 80-Col Plain Paper Printer	695.00
2001-16KN	975.00	2040 Dual Mini Disk Drive	1250.00
2001-16KB (reg)	975.00	Pet to IEEE Cable	39.95
2001-32KN	1250.00	IEEE to IEEE Cable	49.95
2001-32KB (reg)	1250.00	C2N External Cassette Deck	95.00
2022 80-Col Dot Marix Printer	795.00		

CompuCruise™
Zemco, Inc.

Model 44 with cruise control **$174.95**

Model 41 without cruise control **$154.95**

An onboard navigational computer for automobiles, trucks, and recreational vehicles. Features cruise control, fuel management system, trip computer, multifunctioned quartz crystal time counter, plus many other functions.

CRAIG LANGUAGE TRANSLATOR AND INFORMATION CENTER

An instant translator of words and phrases from the world's major languages plus a four function calculator plus information center for nutrition, bar and wine guide, etc.

SPECIAL LOW PRICE LIMITED TIME ONLY (UNTIL AUGUST 17) **$119.95**

PLUS $20.00 REBATE FROM CRAIG
CRAIG MODULES AVAILABLE FROM $24.95

HOME BURGLAR ALARM

GUARDIAN ELECTRONICS, INC.

ACP PRICE ONLY
$189.00

- No installation
- Protects a whole house
- Turns on lights automatically
- Powerful electric siren
- Exit and entry delay
- Battery back-up

The thermostat that uses microprocessor technology to save fuel and money.

PROGRAMMABLE TEMPERATURE CONTROLLER

ACP PRICE
$69.95

A fully automatic electronic thermostat. Easy to install and operate. Compare the cost: TPI's temperature controller is the lowest priced electronic thermostat on the market.

SINGLE BOARD COMPUTER SELECTION GUIDE

BOARD	PROCESSOR	ACP PRICE	ENCLOSURE
KIM-1	6502	169.00	Add 29.95
SYM-1	6502	239.00	Add 39.95
Cromemco	Z80	409.00	N/A
SD-SBC100	Z80	239.00	N/A
AIM 65	6502	375.00	Add 49.95
Cosmac Vip	1802	199.00	Inc.

RETAIL STORES OPEN MON-SAT
STORE #1 1310 "B" E. Edinger Santa Ana, CA 92705 Showrooms, Retail, Warehouse
STORE #2 674 El Camino Real Tustin, CA 92680 Specializing in Systems

P.O. Box 17329 Irvine, Calif. 92713
Direct Order Lines: (714) 558-8813
(800) 854-8230 or (800) 854-8241

FOR INTERNATIONAL ORDERS:
1401 E. Borchard (714) 953-0604
Santa Ana, CA 92705 TWX: 910-595-1565

ADVANCED COMPUTER PRODUCTS

GET YOUR 1980 CATALOG

FIRST TO OFFER PRIME PRODUCTS TO THE HOBBYIST AT FAIR PRICES!

1. **Proven Quality** Factory tested products only.
2. **Guaranteed Satisfaction**
3. **Over $1,000,000.00 Inventory**

1980 CATALOG NOW AVAILABLE.

Send $2.00 for your copy of the most complete catalog of computer products. A must for the serious computer user.

STATIC RAM BOARDS

Now less than 1½¢ per bit!

● **S-100 32K** (uses 2114)

ASSEMBLED		Kit	
450ns.	499.00	450ns.	469.00
250ns.	539.00	250ns.	499.00

Bare Board 49.95
Bare Board w/all parts less mem. 99.95

● **S-100 16K** (S-100 Compatible)
- Low.Power
- 2 MHz or 4 MHz
- Assembled & Tested

WOW!

2 MHz.... **$250.00**
4 MHz.... **$265.00**

● **LOGOS I 8K**
ASSEMBLED

450 ns.	149.95	KIT 450ns.	125.95
250ns.	169.95	250ns.	149.95

Bare PC Board w/Data $21.95
"Special Offer" Buy (4) 8K 450ns. Kits $117.00

The VISTA V-80 Disk Drive System
- 23% more storage capacity than TRS-80
- 120 day warranty
- 40 track patch at NO CHARGE

Single drive system........ $ 395.00
Two drive system.......... $ 770.00
Four drive system......... $1450.00
Two drive cable........... $ 20.95
Four drive cable.......... $ 39.95

THE VISTA V-200 FOR EXIDY
Price: Starting as low as **$1199.00**

Item	Storage Capacity	Drive Configuration	Shipping Weight	Price
V200 E-20	400	2 Drive, Single Head	25 lbs.	1199.00
V200 E-22	800	2 Drive, Double Head	25 lbs.	1549.00
V200 E-30	600	3 Drive, Single Head	32lbs.	1524.00
V200 E-32	1.2 MEG	3 Drive, Double Head	32 lbs.	1999.00

ATTENTION VIDEO HOBBYISTS!!!
★ BOX BUILDERS
★ USE AS REMOTE TUNER/TIMER
★ FULL SCHEMATICS AVAILABLE
★ FOR ONLY $5.90! – FREE W/PURCHASE

A Recent Special Purchase Allows Us To Present The Following:

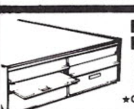
NEW, UNUSED COMPONENTS From The RCA VDT-201 Videocassette Recorder

1. UHF/VHF Tuner Subassembly with all knobs and Video Demodulator **$59.95**
2. RF Modulator with Audio & Video inputs. Channel 3 or 4 Output **$39.95**
3. Digital Clock Module AM/PM Fluorescent Readout (Green) **$19.95**
4. 300 Ohm to 75 Ohm Matching .... **$ 2.49**
or Subtotal **$123.34**
5. Complete Set of All The Above

for Only........ $74.95

Please Call For Volume Discounts

LOW COST FLOPPY DISK SUBSYSTEM



Shugart 801R Drives (2) VISTA Floppy Controller (S-100), Case, Power Supply & Cable. CP/M Disk Operating System Assembled & Tested **$1499.00**
★CHECK OUR FLOPPY DISK PRICING! ON THIS PAGE
SAVE $300.00 ($1799.00 Value)

WATANABE MIPLOT
Max pen speed 50 mm/sec., resolution .10 mm typically .05 mm, built-in character generator sizes and 4 degrees of rotation, absolute draw, relative draw, alpha printing done with simple commands. **$1195.00**

IMS STATIC RAM BOARDS

	250 ns.	450 ns.
8K Static	$209.00	$189.00
16K Static	$449.00	$399.00
32K Static	$729.00	$629.00

ANADEX PRINTER NEW APPLE VERSION
Model DP-8000 compact, impact, parallel or serial. Sprocket feed, 80 cols, 84 lines/min., bi-directional.
New only $875.00
DP-8000AP (for Apple) $875.00

SALE! SD SYSTEMS BOARDS
TAKE 10% OFF!

	KIT	ASSEM
SBC 100 Single Board Computer (2MHz)	$265.00	$ 349.00
SBC 200 Single Board Computer (4MHz)	299.00	399.00
Z80 Starter System	319.00	449.00
VDB 8024 Video Display Board	335.00	459.00
Versa-Floppy II	325.00	429.00
Expando PROM	115.00	225.00
SD100 Computer System w/64K	–	6995.00
SD200 Computer System w/64K	–	7995.00

6800 MICROMODULE™ PRICE LIST

MODEL NO.	DESCRIPTION	PRICE
9600A	Single Board Microcomputer	$495.00
9609	Advanced Single Bd. Comp. (6809)	595.00
9601	16 Slot Mother Board	175.00
9602	Card Cage	75.00
9603	8 Slot Mother Board	100.00
9604	Power Supply	275.00
9605	DC Input Power Supply	325.00
9610	Utility Proto Board	39.00
9611	Arith. Proc/Memory Module	495.00
9612	Buffered Utility Proto Board	49.00
9616	32K EPROM/RAM Module	250.00
9617	EPROM Programming Head	250.00
9620	16 Channel Parallel I/O Module	295.00
9622	Serial-Parallel I/O Module	325.00
9627	16K Static RAM Module 470ns	395.00
9629	32K Static RAM 450ns	695.00
9629A	32K Static RAM 200ns	895.00
9630	Card Extender	68.00
9640	Multiple Programmable Timer	395.00
9650	8 Channel Duplex Serial I/O Mod.	395.00
9655	Intelligent Tape Controller	550.00
96103	32/32 I/O Module	275.00
96702	Contact Closure Module	350.00

UNPOPULATED BOARDS (Also Available)

APPLE/EXIDY/EXPANDO TRS 80 16K-UPGRADE KIT
$54.95 TRS-80/APPLE $54.95

MEMORY EXPANSION KITS, 4116's, 16K (200/250 ns.)
8 pcs for **$54.95**
w/instructions & jumpers
Call For Volume Pricing

★ Special: TRS80 Schematic........ $ 4.95
★ Expansion Interface Schematic.. $ 4.95
★ Expansion Interface Connector... 7.95

EXPANDORAM II MEMORY KITS
★ Bank Selectable ★ Uses 4116 200 ns.
★ Write Protect ★ Power 8VDC, ±16VDC
★ Phantom NEW LOW PRICES ★ Up to 4 MHz

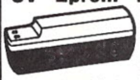
Expando 64 Kit (4116) Assem. & Tested Add $50.

16K	$269.00	48K	$435.00
32K	$349.00	64K	$505.00

HAZELTINE TERMINALS
SALE $749.00

Model 1400... $749.00	Model 1500... $1085.00
Model 1410... $825.00	Model 1510... $1245.00
Model 1420... $945.00	Model 1520... $1495.00

UV "Eprom" Eraser



Model UVs-11E $69.95
Holds 4 Eprom's at a time. Backed by 45 years experience.
Model S-52T... $265.00
Professional Industrial Model

EMAKO-20.. Reg. $777.00 $599.00
UNBELIEVABLE!!
125 Caps, 60 lpm - Vertical Formal Unit - 96 Characters - Upper/Lower Case - 4.5" to 9.5" Adjustable - 80 col/40 col double width - Full 96 char. ASCII

EMAKO-22............$799.00
Prints a 132 col/line. Available with parallel or serial output at same price.

MIKA 20.............$1280.00
9x7, 125 cps 136 characters/line
Full 15" width. Super for business applications requiring large IBM format paper.

BASE II PRINTER
80 Column Impact Printer
- 60 Lines Per Minute
- 115/230 VAC, 50 or 60 Hz.
- 72, 80, 96, 120 or 132 Char/Line
- Self-Test Switch

REG. $649.00
ACP PRICE **$550.00**

Option "M" Terminal Screen Buffer (1920 Char.)........ $50.00
Option "S" High Speed Paper Advance & Graphics....... 50.00
Option "T" Tractor Feed....... 50.00

Z-80/Z-80A/8080 CPU BOARD
★ On board 2708 ★ 2708 included (450ns.)
★ Power on jump ★ completely socketed
● Z-80 Assembled and Tested $185.00
● Z-80 Kit.......... $129.95
● Z-80 Bare PC Board .... $ 34.95
★ For 4MHz Speed Add $15.00
8080A Kit .... $ 99.95
8080A Assembled.......... $149.95

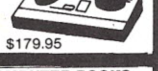
S-100 MOTHERBOARD SPECIAL
8 slot expandable w/9 conn.
reg $69.95.......... NOW $52.95

SIEMEN'S FLOPPY
- Special buy while supply lasts.
- 8" Drive with Double-Density
- 90 Day Warranty

★CHECK OUR FLOPPY DISK PRICING! ON THIS PAGE

SALE
SIEMENS $429.00
SHUGART 801R $475.00

ACOUSTIC MODEM
NOVATION CAT™
0-300 Baud
Bell 103
Answer, Originate $179.95



DATA BOOKS • COMPUTER BOOKS

1980 IC Master 59.95	Intel MCS 80 Manual......7.95
NSC TTL Data.......3.95	Intel MCS 40 Manual......4.95
NSC Linear.......4.95	AMD 8080A Manual.......5.95
NSC Linear App Notes II . 3.95	AMD Schottky Databook .. 4.95
NSC CMOS.......3.95	AMI MOS/LSI Data.......3.95
NSC Memory.......3.95	GI MOS/LSI Data 4.95
Intel Databook 7.50	Harris Analog Databook... 4.95
Intel MCS 85 Manual.....7.50	TI Linear Control Data.....3.95

SALE • OSBORNE BOOKS • SALE

	Reg.	Sale
Intro to Micros Vol. 0	~~8.50~~	7.75
Intro to Micros Vol. I	~~8.50~~	7.75
8080A Programming	~~8.50~~	7.75
6800 Programming	~~8.50~~	7.75
Z80 Programming	~~8.50~~	7.75
Vol. II Some Real Microprocessors w/Binder	~~30.00~~	27.50
Vol. III Some Real Support Devices w/Binder	~~20.00~~	18.50
Intro to Micros Vol. III	~~20.00~~	18.50

SALE • DILITHIUM COMPUTER BOOKS • SALE

Understanding Computers	~~8.95~~	7.95
8080Microcomputer Experiments	~~13.95~~	11.95
Beginning BASIC	~~9.95~~	8.95
Beginners Glossary & Guide	~~6.95~~	5.95
Peanut Butter & Jelly Guide to Computers	~~7.95~~	6.95
8080 Machine Language Programming	~~7.95~~	6.95
Home Computers Vol. I Hardware	~~7.95~~	6.95
Home Computers Vol. II Software	~~13.95~~	11.95
Starship Simulator	~~7.95~~	6.95

MICROPROCESSORS

Z8001 16 bit to 8Mb....... $189.00
Z8002 16 bit to 64K....... 149.00
Z80....... 10.75
Z80A....... 14.50
F-8 (3850)....... 16.95
2650....... 18.95
CD1802....... 13.95
8080A....... 8.50
8080A-4MHz....... 19.95
SALE 8085....... 19.95
8008-1....... 14.95
2901....... 9.90
2901A....... 14.95
2903 4-bit Superslice....... 29.95
TMS 9900JL....... 49.95
CP1600....... 39.95
6502....... 11.50
6502A....... 16.95
IM6100....... 29.95
6800....... 11.75
6800B 2.0 MHz....... 19.95
6802P....... 17.95
8035....... 19.95
8755....... 49.95
8748....... 69.95
6809....... 64.95
8086....... 69.95

ADVANCED SUPPORT
AM9511 Arith. Processor ... 175.00
9512 Arith Processor....... 175.00
9513 Univ. Timing....... 79.95
AM9517 DMA Controller 18.95
AM9519 Universal Interrupt.. 18.95

Z-80 SUPPORT CHIPS
Z80-PIO 2.5 MHz....... 8.75
Z80A-PIO 4.0 MHz....... 12.95
Z80-CTC 2.5 MHz....... 8.75
Z80A-CTC 4.0 MHz....... 12.95
Z80-DMA 2.5 MHz....... 29.95
Z80A-DMA 4.0 MHz....... 36.95
Z80-SIO/0 2.5 MHz....... 35.95
Z80A-SIO/0 4.0 MHZ....... 39.40
Z80-SIO/1 2.5 MHz....... 35.95
Z80A-SIO/1 4.0 MHz....... 39.40
Z80-SIO/2 2.5 MHz....... 35.95
Z80-SIO/2 4.0 MHz....... 39.40

8080/8085 SUPPORT
8155/8156 I/O....... 24.95
8755 I/O with Eprom....... 64.95
8202 Dyn. Ram Cont....... 34.95
8205/74S 138 Decoder....... 3.95
8212 8 bit I/o....... 2.75
8214 Priority Int....... 5.25
8216 Bus Driver....... 2.75
8224 Clock Gen....... 2.95
8224-4 (4MHz)....... 9.75
8226 Bus Driver....... 3.95
8T26 Bus Driver....... 2.95
8228 Sys. Control....... 5.50
8238 Sys. Cont....... 5.50
8251 Prog. I/O....... 6.95
8253 Int. Timer....... 16.95
8255 Prog. I/O....... 6.50
8257 Prog. DMA....... 16.95
8259 Prog. Int....... 17.95
8275 CRT Controller....... 59.95
8279 Prog. Keyboard....... 18.95

6800 SUPPORT CHIPS
6810 128 x 8 Ram....... 4.75
6820 PIA....... 5.95
6821 PIA....... 6.50
6828 Priority Int....... 9.95
6834-1 512 x 8 Eprom....... 16.95
6845/HD46505 CRT Cont....... 39.95
6847 Color CRT....... 49.95
6850 ACIA....... 5.95
6852 Serial Adapter....... 5.95
6860 Modem....... 10.95
6862 Modulator....... 11.95
6871A 1.0MHz OSC....... 25.95
6875....... 8.25
6880 Bus Driver....... 2.85
MC68488....... 19.95
68047....... 24.95

1802 SUPPORT CHIPS
1821 SCD 2K RAM....... 25.00
1822 SCD 256 x 4 RAM....... 15.95
1824 CD 32 x 8 RAM....... 9.95
1852 CD 8 bit I/o....... 10.95
1854 Uart....... 10.95
1856 CD I/O....... 8.95
1858 CD I/O....... 8.95
1861....... 12.95

6502 SUPPORT CHIPS
6520 PIA....... 7.50
6522 Mult....... 11.95
6530-002,003,004,005....... 21.95
6532....... 19.95
6551....... 19.95

PROMS
2708 450 ns....... 8.25
2708-6 650 ns....... 7.50
1702A....... 4.95
2732....... 74.95
2716-5V....... 29.95
2716-5V, 12V....... 29.95
2758-5V....... 29.95
5203AQ....... 13.95
5204AQ....... 14.95
IM 5610....... 3.90
SALE 8223 32 x 8....... 2.95
82S115 512 x 8 (TS)....... 16.95
82S123 32 x 8....... 4.90
82S126 256 x 4....... 4.90
82S129 256 x 4 (TS)....... 4.90
82S130 512 x 4 (OC)....... 6.50
82S23....... 6.95
82S131....... 14.95
82S137....... 14.95
NOTE: WE PROGRAM PROMS

CHARACTER GEN.
2513-001 (5V) Upper....... 9.50
2513-005 (5V) Lower....... 10.95
2513-ADM3 (5V) Lower....... 14.95
MCM6571....... 11.75
MCM6571A....... 11.75
MCM6574....... 14.50
MCM6575....... 14.50

UARTS/BAUD RATE
TR1602B (5V, 12V)....... 3.95
AY51013 (5V, 12V)....... 4.95
AY51014A/1612 (5-14V)....... 6.95
AY51015A/1863 (5V)....... 6.95
TMS 6011 (5V, 12V)....... 5.50
IM6402....... 7.95
IM6403....... 8.95
2350 USRT....... 9.95
1671B Astros....... 24.95
SALE TR1472B....... 9.95
MC14411....... 11.95
4702....... 14.95
WD1941....... 9.95
COM 5016....... 16.95

KEYBOARD ENCODERS
AY5-2376....... 13.75
AY5-3600....... 13.75
HD0165....... 9.95
74C922....... 9.95
74C923....... 9.95

STATIC RAMS

	1-24	25-99	100
21L02 450ns.	1.30	1.25	.99
21L02 250ns.	1.59	1.55	1.15
2111	3.75	3.65	3.55
2112-1	2.95	2.85	2.65
2101-1	2.90	2.70	2.55
2114L-250ns. (4045)	8.95	7.50	6.50
2114L-450ns. (4045)	5.50	5.25	4.75
4044 250 ns.	8.95	7.50	6.90
4044 450ns.	5.95	5.95	4.75
EMM4200A	9.75	8.75	7.95
EMM4402	7.95	7.25	6.25
EMM 4804	12.50	11.50	9.95
5101C-E	7.95	7.95	7.25
AMD9140/41	10.95	10.25	9.25
AMD9130/31	12.95	11.95	10.25
1101	1.95	1.75	1.25
P2125/93425 (45ns.)	9.95	8.35	8.25
6508 1K x 1 CMOS	7.95	7.95	7.25
6518 1K x 1 CMOS	7.95	7.95	7.25
74S189 64 bit Ram	3.95	3.25	2.50
2147 Low Power 4K Static	19.95	18.95	16.95

DYNAMIC RAMS
416/4116 16K (16 Pin)....... 8.75
Set of 8 416's....... 54.95
4115 8K (16 Pin)....... 6.95
4050 4K x 1 (18 Pin)....... 4.95
4060 4K x 1 (22 Pin)....... 4.95
4096 4K x 1 (16 Pin)....... 3.95
2104 4K x 1 (16 Pin)....... 4.75
4027 4K x 1 (16 Pin)....... 4.95

64K RAMS

5261	1.95	1103	1.95
5262	1.95	4008L	4.95
5270	4.95	6605	7.95
5280	4.95	6604	4.95
5290	12.45	6002	1.50

$175.00

SOCKETS

# Pins	Lo-Pro Soldertail	Wire Wrap 3 Level Tin	Wire Wrap 3 Level Gold
8	.15	.32	.45
14	.19	.36	.59
16	.20	.38	.62
18	.24	.59	.84
20	.29	.69	.99
22	.24	.79	1.10
24	.38	.85	1.20
28	.43	1.10	1.49
36	.58	1.25	1.69
40	.60	1.40	1.89

ZERO INSERTION FORCE
16 Pin $5.50 24 Pin $7.50 40 Pin $10.25

LED READOUTS

Part No.	Size	Color	Description	Price
DL704	.300	Red	Comm. Cathode	.99
DL707	.300	Red	Comm. Anode	.99
FND357	.357	Red	Comm. Cathode	.99
FND501	.500	Red	Comm. Cathode (+1)	.99
FND503 (500)	.500	Red	Comm. Cathode	.99
FND507 (510)	.500	Red	Comm. Anode	.99
FND508	.500	Red	Comm. Anode (+1)	.99
FND530	.500	Green	Comm. Cathode	1.49
FND550	.500	Orange	Comm. Cathode	1.49
FND803 (800)	.800	Red	Comm. Cathode	1.75
FND807 (810)	.800	Red	Comm. Anode	1.75
HP5082-7340	.600	Red	4 x 7 Hexidecimal	20.95
HP5082-7300	.600	Red	4 x 7 Sgl. Digit RHD	18.95
HP5082-7731	.300	Red	Comm. Anode	.75
TIL305	–	Red	Array 5 x 7	7.95
TIL308	.270	Red	Numerical Display	10.95
TIL309	.270	Red	Numerical Display	9.95
TIL311	.270	Red	4 x 7 Hexidecimal	10.95
MAN2A	.320	Red	Array	5.95
MAN10A	.270	Red	Alpha Numeric	8.95
XAN3061	.300	Red	Comm. Anode Right DP	1.25
XAN3062	.300	Red	Comm. Anode Left DP	1.25
XAN3063	.300	Red	Overflow, CA, Left DP	1.25
XAN3064	.300	Red	Comm. Cathode Right DP	1.25
XAN3051	.300	Green	Comm. Anode Right DP	1.99
XAN3052	.300	Green	Comm. Anode Left DP	1.99
XAN3053	.300	Green	Overflow CA, Left DP	1.99
XAN3054	.300	Green	Comm. Cathode, Right DP	1.99
XAN3081	.300	Yellow	Comm. Anode, Right DP	2.10
XAN3082	.300	Yellow	Comm. Anode, Left DP	2.10
XAN3083	.300	Yellow	Overflow CA, Left DP	2.10
XAN3084	.300	Yellow	Comm. Cathode, Right DP	2.10

IC SPECIAL PURCHASE
LF 13508 JFET Analog Multi 8 bit....... 8.95
ICM 7045 Precision StopWatch....... 23.95
ICM 7205 CMOS LED Stopwatch/Timer.. 19.95
ICM 7207 Oscillator Controller....... 7.50
ICM 7208 Seven Decade Counter....... 18.95
ICM 7209 Clock Generator....... 6.95
ICL 7107 3½ Digit A/D (LED)....... 14.95
MC14433P 3½ Digit A/D Converter....... 13.95
ICL 8211 Voltage Reference....... 1.95
LM 390N Battery Op. Audio Amp....... 3/1.00
LM 1800N PLL FM Stereo....... 3/1.00
LM 1820N AM Radio....... 3/1.00
LM 1850N Ground Fault IC....... 3/1.00
LM 2900N Quad Amplifier....... 2/1.00
LM 2901N (HM) Quad Comparator....... 2/1.00
LM 2917N Freq. to Volt Conv....... 2/1.99
AY-3-3550 4½ digit DMM....... 24.95
AY-5-3507 40MHz DVM....... 9.95
MEM 4963 Mosfet Smoke Detector 11.95
27S08 32 x 8 Prom (IM 5600)....... 2.95
G115M 6 channel Mosfet....... 1.49
ULN 2003A 7 channel Driver....... .99
10146 1K x 1 ECL Ram....... 9.95
MCM 14505 64 bit Static Ram....... 6.95

7520/25	.55	LM 377	1.25
1488/1489	2/1.99	LM 387	1.25
MK5014 Calc.	3/1.99	LM 3302	3/1.99
74589	3/1.99	LM 4558	3/1.99
78HO5K	6.95	RC 4136	2/1.99
LM323K	4.95	RC 4131	2/1.99
MC1372	8.95	COM 5027	39.95
		COM 5037	49.95

VERBATIM DISKETTES
100% CERTIFIED ERROR-FREE!

● 35 TRACKS, DOUBLE DENSITY, SINGLE-SIDED 5¼

TYPE	DESCRIPTION	PRICE/BOX 10
525-01	Soft Sectored TRS 80, etc.	$27.25
525-10	10 Hole Hard NS, Apple, etc.	27.25
525-16	16 Hole Hard, Micropolis	27.25

● 40 & 77 TRACK DOUBLE DENSITY – SINGLE SIDED 5¼" W/HUB REINFORCEMENT RING

577-01	Soft Sectored, TRS 80, etc.	$32.25
577-10	10 Hole Hard NS, Apple, etc.	32.25
577-16	16 Hole Hard, Micropolis	32.25

● 40 TRACKS, DOUBLE SIDED, DOUBLE DENSITY 5¼

550-01	Soft Sectored SA 450, MPI 52	$39.50
550-10	10 Hole Hard, BASF, WANGCO	39.50
550-16	16 Hole Hard, Micro²	39.50

● 8" 100% CERTIFIED DISKETTES

FD800-01	Soft, Single Density, 3740	$33.50
FD800-32	32 Hard Single Density Shug.	33.50
FD802-01	Soft, Double Density, 3740	44.80
FD850-01	Soft, Double Sided, Double Density, 3740	49.80

VOLUME DEALER PRICING AVAILABLE
WE ALSO STOCK DYSAN-CALL

FLOPPY DISK I/O
1771-01 8" & Minifloppy....... 24.95
uPd372 Nec Floppy....... 49.95
1781 Dual Floppy....... 29.95
1791-01 Dual Floppy....... 36.95
uPd 765 Floppy....... 49.95

A/D CONVERTERS
8700 8 bit Binary....... 13.50
8701 10 bit Binary....... 22.00
8703 8 bit TS....... 13.50
9400 Volt to Freq. Conv....... 7.25
8750 3½ Digit BCD....... 13.95
1408L6 6 bit....... 3.95
1408L8 8 bit....... 5.95
DACOI D to A....... 5.95

TV CHIPS/SOUND
AY38500-1 6 Games B/W....... 4.95
AY38515 Color Converter....... 3.95
AY38603-1 Roadrace Game....... 8.95
AY38605-1 Warfare Game....... CALL
AY38606-1 Wipeout Game....... 9.50
AY38607-1 Shooting Gallery....... 8.95
AY38910 Sound Generator....... 12.95
SN76477 TI Sound Generator....... 3.95
MM5320/21 TV Synch Gen....... 9.95
MM5369 Prescaler....... 3.95
LM1889 RF Modulator....... 3.95
MM57100 NSC Color TV....... 6.95
MM57104 Clock Gen....... 3.75
RF Modulator w/Audio....... 8.95
M&R Modulator....... 29.95

WAVEFORM GEN.
8038 Function Gen....... 3.95
MC4024VCO....... 2.95
LM566 VCO....... 1.95
XR2206 Function Generator....... 5.25

SHIFT REGISTERS
MM500H Dual25....... .50
MM5056N Dual 256....... 2.95
MM5060N Dual 128....... 2.95
2510A Dual 100....... 1.95
2847 Quad 80....... 4.95
3341 Dual 80....... 4.95
3351 40 x 9 FIFO....... 17.95
3357 Quad 80....... 6.95
9403 16 x 4 FIFO....... 24.95
9408 10 Bit Sequencer....... 9.95

CTS DIPSWITCHES

CTS206-2	1.75	CTS206-7	1.75
CTS206-4	1.75	CTS206-8	1.95
CTS206-5	1.75	CTS206-9	1.95
CTS206-6	1.75	CTS206-10	1.95

CONNECTORS (GOLD)
DB25P (RS232)....... 3.25
DB25S Female....... 3.75
Hood....... 1.26
Set w/Hood, Sale....... $7.50
22/44 W/W, S/T, KIM....... 2.95
43/86 W/W, S/T, MOT....... 6.50
50/100 S-100 Connector w/w....... 4.95
50/100 S-100 Connector s/t....... 3.95

NAKED PC
Z-80 CPU (Ithaca)....... $34.95
8080A CPU....... 34.95
8K Static RAM (Logos)....... 24.95
16K Static RAM (2114)....... 29.95
32K Static RAM (2114)....... 29.95
8K Eprom (2708)....... 24.95
1702 Eprom Board....... 30.00
2708/2716 Eprom (Ithaca)....... 34.95
2708/2716 Eprom (WMC)....... 30.00
Realtime Clock....... 34.95
ACP Proto Bd. (3M Conn.)....... 27.95
Vector 8800 Proto....... 22.20
Vector 8803 11 slot MB....... 29.95
ACP Extender w/Conn....... 18.95
Video Interface (SSM)....... 31.95
Parallel Interface (SSM)....... 31.95
13 Slot MotherBoard (WMC) 32.95
9 Slot MotherBoard (WMC)....... 29.95
8 Slot Mother (expandable)....... 34.95
Proc-Tech Bare Boards....... CALL

COMPUTER SPECIALS

DISCOUNTS!	LIST	SALE
Apple II Plus w/16K	1195.	990.
PET 2001-16N	995.	895.
Exidy Sorcerer w/16K		1099.
Cromemco Sys III	6990.	6290.
Horizon I w/32K		CALL
Pascal Microengine	1995.	1595.
IPSI 1620 Diablo RO	3295.	2695.
Anadex DP 8000	995.	875.
Centronics Micro S-1	595.	525.
Soroc IQ 120	995.	795.
Teletype Model 43	1349.	1150.
HiPlot Plotter	1085.	899.
HiPlot Digitizer	795.	735.
Intertube II	895.	784.
Atari 800	999.	750.
Atari 400	549.	449.
TI 99/4	1150.	1099.
Leedex 12" Monitor	149.	139.
Centronics 737	995.	899.
Trencom T-100	375.	349.
Trencom T-200	595.	525.
Sanyo 9" Monitor	220.	169.
Sanyo 15" Monitor	295.	249.
Mot 22 MHz Monitor	296.	219.

FREE CASSETTE
with purchase of Atari 800 or Atari 400, receive any available Programma cassette worth $19.95

SOCKET SPECIALS
14 Pin w/w-2L....... $ .35
16 Pin w/w-2L....... .40
18 Pin w/w-2L....... .55
20 Pin w/w-2L....... .75
40 Pin w/w-2L....... 1.45
18 Pin s/t....... .15
22 Pin s/t....... .15
Call for volume pricing.

TERMS: Use check, M/C, VISA, AMEX, CB, or COD. COD requires 25% deposit. Charge Orders please include expiration date. Foreign pay U.S. Funds. Order by phone, mail or TWX. MINIMUM $10.00. Please include magazine issue and phone no. SHIPPING: U.S.A. add $2.00 for first 2 lbs. For surface add 30¢ for additional lbs. For air add 70¢ for additional lbs. FOREIGN: Add 10% shipping and handling. COD's $1.85 extra. Not responsible for typos. Some items subject to prior sale. We reserve right to limit quantities, some items subject to precious metal adder. Retail pricing may vary from Mail Order. We reserve the right to substitute manufacturer.

RETAIL STORES OPEN MON-SAT
STORE #1 1310 "B" E. Edinger Santa Ana, CA 92705 Showrooms, Retail, Warehouse
STORE #2 674 El Camino Real Tustin, CA 92680 Specializing in Systems

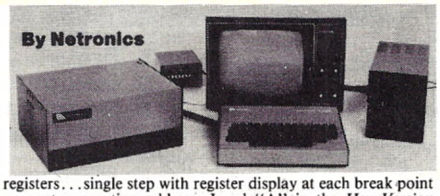
P.O. Box 17329 Irvine, Calif. 92713
Direct Order Lines: (714) 558-8813
(800) 854-8230 or (800) 854-8241

FOR INTERNATIONAL ORDERS:
1401 E. Borchard (714) 953-0604
Santa Ana, CA92705 TWX: 910-595-1565

Start Computing For Just $129.95 With An 8085-Based Professional Computer Kit—

Explorer/85

100% compatible with all 8080A and 8085 software & development tools!



No matter what your future computing plans may be, Level "A"—at $129.95—is your starting point.

Starting at just $129.95 for a Level "A" operating system, you can now build the exact computer you want. Explorer/85 can be your beginner's system, OEM controller, or IBM-formatted 8" disk small business system... yet you're never forced to spend a penny for a component or feature you don't want and you can expand in small, affordable steps!

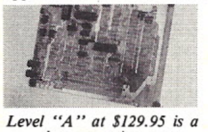
Now, for just $129.95, you can own the first level of a fully expandable computer with professional capabilities—a computer which features the advanced Intel 8085 cpu, *thereby giving you immediate access to all software and development tools that exist for both the 8085 and its 8080A predecessor (they are 100% software compatible)*—a computer which features onboard S-100 bus expansion—plus instant conversion to mass storage disk memory with either 5-1/4" diskettes or standard IBM-formatted 8" disks.

For just $129.95 (plus the cost of a power supply, keyboard/terminal and RF modulator, if you don't have them already), Explorer/85 lets you begin *computing* on a significant level... *applying* the principles discussed in leading computer magazines... *developing* "state of the art" computer solutions for both the industrial and leisure environment.

Level "A" Specifications

Explorer/85's Level "A" system features the advanced Intel 8085 cpu, an 8355 ROM with 2k deluxe monitor/operating system, and an 8155 ROM-I/O—all on a single motherboard with room for RAM/ROM/PROM/EPROM and S-100 expansion, *plus* generous prototyping space.

(Level "A" makes a perfect OEM controller for industrial applications and is available in a special Hex Version which can be programmed using the Netronics Hex Keypad/Display.)



Level "A" at $129.95 is a complete operating system, perfect for beginners, hobbiests, or industrial controller use.

PC Board: glass epoxy, plated through holes with solder mask • **I/O:** provisions for 25-pin (DB25) connector for terminal serial I/O, which can also support a paper tape reader ...provision for 24-pin DIP socket for hex keyboard/display... cassette tape recorder input... cassette tape recorder output... cassette tape control output... speaker output... LED output indicator on SOD (serial output) line... printer interface (less drivers)... total of four 8-bit plus one 6-bit I/O ports • **Crystal Frequency:** 6.144 MHz • **Control Switches:** reset and user (RST 7.5) interrupt... additional provisions for RST 5.5, 6.5 and TRAP interrupts onboard • **Counter/Timer:** programmable, 14-bit binary • **System RAM:** 256 bytes located at F800, ideal for smaller systems and for use as an isolated stack area in expanded systems... RAM expandable to 64k via S-100 bus or 4K on motherboard.

System Monitor (Terminal Version): 2k bytes of deluxe system monitor ROM located at F000 leaving 0000 free for user RAM/ROM. Features include tape load with labeling ... tape dump with labeling... examine/change contents of memory ... insert data... warm start... examine and change all registers... single step with register display at each break point, a debugging/training feature... go to execution address... move blocks of memory from one location to another... fill blocks of memory with a constant... display blocks of memory ... automatic baud rate selection... variable display line length control (1-255 characters/line)... channelized I/O monitor routine with 8-bit parallel output for high speed printer... serial console in and console out channel so that monitor can communicate with I/O ports.

System Monitor (Hex Version): Tape load with labeling... tape dump with labeling... examine/change contents of memory... insert data... warm start... examine and change all registers... single step with register display at each break point ...go to execution address. Level "A" in the *Hex Version* makes a perfect controller for industrial applications and can be programmed using the Netronics Hex Keypad/Display.

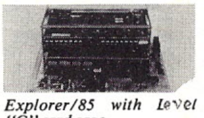
Hex Keypad/Display.

Hex Keypad/Display Specifications

Calculator type keypad with 24 system defined and 16 user defined keys. 6 digit calculator type display which displays full address plus data as well as register and status information.

Level "B" Specifications

Level "B" provides the S-100 signals plus buffers/drivers to support up to six S-100 bus boards and includes: address decoding for onboard 4k RAM expansion select-able in 4k blocks... address decoding for onboard 8k EPROM expansion selectable in 8k blocks... address and data bus drivers for onboard expansion... wait state generator (jumper selectable), to allow the use of slower memories... two separate 5 volt regulators.



Explorer/85 with Level "C" card cage.

Level "C" Specifications

Level "C" expands Explorer's motherboard with a card cage, allowing you to plug up to six S-100 cards directly into the motherboard. Both cage and cards are neatly contained inside Explorer's deluxe steel cabinet. Level "C" includes a sheet metal superstructure, a 5-card gold plated S-100 extension PC board which plugs into the motherboard. Just add required number of S-100 connectors

Level "D" Specifications

Level "D" provides 4k or RAM, power supply regulation, filtering decoupling components and sockets to expand your Explorer/85 memory to 4k (plus the original 256 bytes located in the 8155A). The static RAM can be located anywhere from 0000 to EFFF in 4k blocks.

Level "E" Specifications

Level "E" adds sockets for 8k of EPROM to use the popular Intel 2716 or the TI 2516. It includes all sockets, power supply regulator, heat sink, filtering and decoupling components. Sockets may also be used for soon to be available RAM IC's (allowing for up to 12k of onboard RAM).

Order A Coordinated Explorer/85 Applications Pak!

Experimenter's Pak (SAVE $12.50)—Buy Level "A" and Hex Keypad/Display for **$199.90** and get FREE Intel 8085 user's manual plus FREE postage & handling!

Student Pak (SAVE $24.45)—Buy Level "A," ASCII Keyboard/Computer Terminal, and Power Supply for **$319.85** and get FREE RF Modulator plus FREE Intel 8085 user's manual plus FREE postage & handling!

Engineering Pak (SAVE $41.00)—Buy Levels "A," "B," "C," "D," and "E" with Power Supply, ASCII Keyboard/Computer Terminal, and six S-100 Bus Connectors for **$514.75** and get 10 FREE computer grade cassette tapes plus FREE 8085 user's manual plus FREE postage & handling!

Business Pak (SAVE $89.95)—Buy Explorer/85 Levels "A," "B," and "C" (with cabinet), Power Supply, ASCII Keyboard/Computer Terminal (with cabinet), 16k RAM, 12" Video Monitor, North Star 5-1/4" Disk Drive (includes North Star BASIC) with power supply and cabinet, all for just **$1599.40** and get 10 FREE 5-1/4" minidiskettes ($49.95 value) plus FREE 8085 user's manual plus FREE postage & handling!

Continental U.S.A. Credit Card Buyers Outside Connecticut

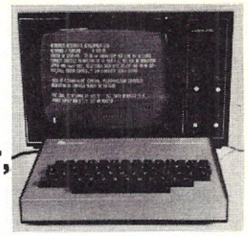
CALL TOLL FREE 800-243-7428

To Order From Connecticut Or For Technical Assistance, Etc. **Call (203) 354-9375**

Netronics R&D Ltd.
333 Litchfield Road, New Milford, CT 06776
Please send the items checked below—

- ☐ **Explorer/85 Level "A" Kit** (ASCII Version), **$129.95** plus $3 p&h.
- ☐ **Explorer/85 Level "A" Kit** (Hex Version), **$129.95** plus $3 p&h.
- ☐ **8k Microsoft BASIC** on cassette tape, **$64.95** postpaid.
- ☐ **8k Microsoft BASIC in ROM Kit** (requires Levels "B," "D," and "E"), **$99.95** plus $2 p&h.
- ☐ **Level "B" (S-100) Kit, $49.95** plus $2 p&h.
- ☐ **Level "C" (S-100 6-card expander) Kit, $39.95** plus $2 p&h.
- ☐ **Level "D" (4k RAM) Kit, $69.95** plus $2 p&h.
- ☐ **Level "E" (EPROM/ROM) Kit, $5.95** plus 50¢ p&h.
- ☐ **Deluxe Steel Cabinet** for Explorer/85, **$49.95** plus $3 p&h.
- ☐ **ASCII Keyboard/Computer Terminal Kit** (features a full 128 character set, upper & lower case, full cursor control, 75 ohm video output convertible to baudot output, selectable baud rate, RS232-C or 20 ma. I/O, 32 or 64 character by 16 line formats, and can be used with either a CRT monitor or a TV set (if you have an RF modulator), **$149.95** plus $2.50 p&h.
- ☐ **Hex Keypad/Display Kit, $69.95** plus $2 p&h.
- ☐ **Deluxe Steel Cabinet** for ASCII Keyboard/Terminal, **$19.95** plus $2.50 p&h.
- ☐ **Power Supply Kit** (±8V @ 5 amps) in deluxe steel cabinet, **$39.95** plus $2 p&h.
- ☐ **Gold Plated S-100 Bus Connectors, $4.85** each, postpaid.
- ☐ **RF Modulator Kit** (allows you to use your TV set as a monitor), **$8.95** postpaid.
- ☐ **16k RAM Kit** (S-100 Board expands to 64k), **$199.95** plus $2 p&h.
- ☐ **32k RAM Kit, $329.95** plus $2 p&h.
- ☐ **48K RAM Kit, $459.95** plus $2 p&h.
- ☐ **64k RAM Kit, $589.95** plus $2 p&h.
- ☐ **16k RAM Expansion Kit** (to expand any of the above up to 64k), **$139.95** plus $2 p&h each.
- ☐ **Intel 8085 cpu User's Manual, $7.50** postpaid.
- ☐ **Special Computer Grade Cassette Tapes, $1.90** each or **3 for $5**, postpaid.
- ☐ **12" Video Monitor** (10 MHz bandwidth), **$139.95** plus $5 p&h.
- ☐ **North Star Double Density Floppy Disk Kit** (One Drive) for Explorer/85 (includes 3 drive S-100 controller, DOS, and extended BASIC with personalized disk operating system—just plug it in and you're up and running!), **$699.95** plus $5 p&h.
- ☐ **Power Supply Kit** for North Star Disk Drive, **$39.95** plus $2 p&h.
- ☐ **Deluxe Case** for North Star Disk Drive, **$39.95** plus $2 p&h.
- ☐ **Experimenter's Pak** (see above), **$199.90** postpaid.
- ☐ **Student Pak** (see above), **$319.85** postpaid.
- ☐ **Engineering Pak** (see above), **$514.75** postpaid.
- ☐ **Business Pak** (see above), **$1599.40** postpaid.

Total Enclosed $ ________
(Conn. res. add sales tax) By—
☐ Personal Check ☐ M.O./Cashier's Check ☐ Visa ☐ Master Charge
(Bank # ________)
Acct. # ________
Signature ________ Exp. Date ____
Print Name ________
Address ________
City ________
State ________ Zip ________
☐ **Send Me Information**



By Netronics

ASCII/BAUDOT, STAND ALONE

Computer Terminal

COMPLETE FOR ONLY... $149.95

The Netronics ASCII/BAUDOT Computer Terminal Kit is a microprocessor-controlled, stand alone keyboard/terminal requring no computer memory or software. It allows the use of either a 64 or 32 character by 16 line professional display format with selectable baud rate, RS232-C or 20 ma. output, full cursor control and 75 ohm composite video output.

The keyboard follows the standard typewriter configuration and generates the entire 128 character ASCII upper/lower case set with 96 printable characters. Features include onboard regulators, selectable parity, shift lock key, alpha lock jumper, a drive capability of one TTY load, and the ability to mate directly with almost any computer, including the new Explorer/85 and ELF products by Netronics.

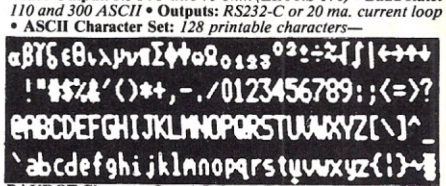
The Computer Terminal requires no I/O mapping and includes 1k of memory, character generator, 2 key rollover, processor controlled cursor control, parallel ASCII/BAUDOT to serial conversion and serial to video processing—fully crystal controlled for superb accuracy. PC boards are the highest quality glass epoxy for the ultimate in reliability and long life.

VIDEO DISPLAY SPECIFICATIONS

The heart of the Netronics Computer Terminal is the microprocessor-controlled Netronics Video Display Board (VID) which allows the terminal to utilize either a parallel ASCII or BAUDOT signal source. The VID converts the parallel data to serial data which is then formatted to either RS232-C or 20 ma. current loop output, which can be connected to the serial I/O on your computer or other interface, i.e., Modem.

When connected to a computer, the computer must echo the character received. This data is received by the VID which processes the information, converting to data to video suitable to be displayed on a TV set (using an RF modulator) or on a video monitor. The VID generates the cursor, horizontal and vertical sync pulses and performs the housekeeping relative to where the character and where it is to be displayed on the screen.

Video Output: *1.5 P/P into 75 ohm (EIA RS-170)* • **Baud Rate:** *110 and 300 ASCII* • **Outputs:** *RS232-C or 20 ma. current loop* • **ASCII Character Set:** *128 printable characters—*



BAUDOT Character Set: *A B C D E F G H I J K L M N O P Q R S T U V W X Y Z - ? : • 3 $ # ( ) . , 9 0 1 4 ! 5 7 ; 2 / 6 8 •* **Cursor Modes:** *Home, Backspace, Horizontal Tab, Line Feed, Vertical Tab, Carriage Return. Two special cursor sequences are provided for absolute and relative X-Y cursor addressing* • **Cursor Control:** *Erase, End of Line, Erase of Screen, Form Feed, Delete* • **Monitor Operation:** *50 or 60Hz (jumper selectable).*

Continental U.S.A. Credit Card Buyers Outside Connecticut

CALL TOLL FREE 800-243-7428

To Order From Connecticut Or For Technical Assistance, Etc. **Call (203) 354-9375**

Netronics R&D Ltd., Dept. K-8
333 Litchfield Road, New Milford, CT 06776
Please send the items checked below—

- ☐ **Netronics Stand Alone ASCII Keyboard/Computer Terminal Kit, $149.95** plus $3.00 postage & handling.
- ☐ **Deluxe Steel Cabinet** for Netronics Keyboard/Terminal In Blue/Black Finish, **$19.95** plus $2.50 postage and handling.
- ☐ **Video Display Board Kit** alone (less keyboard), **$89.95** plus $3 postage & handling.
- ☐ **12" Video Monitor** (10 MHz bandwidth) fully assembled and tested, **$139.95** plus $5 postage and handling.
- ☐ **RF Modulator Kit** (to use your TV set for a monitor), **$8.95** postpaid.
- ☐ **5 amp Power Supply Kit** In Deluxe Steel Cabinet (±8VDC @ 5 amps, plus 6-8 VAC), **$39.95** plus $2 postage & handling.

Total Enclosed (Conn. res. add sales tax) $ ________
By—
☐ Personal Check ☐ Cashiers Check/Money Order
☐ **Visa** ☐ **Master Charge** (Bank # ______)
Acct. # ________
Signature ________ Exp. Date ________
Print Name ________
Address ________
City ________
State ________ Zip ________
☐ **Send Me More Information**


32K STATIC RAM
S-100 MEMORY BOARD
$499.95 ASSEMBLED & TESTED
· FULLY STATIC OPERATION
· 4K BANK ADDRESSABLE
· EXTENDED MEMORY MGMT.
· MEETS IEEE PROPOSED S-100 SIGNAL STANDARDS
· 4 MHZ OPERATION
California Computer Systems
16K STATIC RAM
same features as above.
$249.00
CIRCUITS
74LS00 .38
74LS02 .34
74LS03 .34
74LS04 .38
74LS08 .35
74LS09 .38
74LS10 .28
74LS20 .28
74LS21 .38
74LS22 .38
74LS26 .39
74LS27 .36
74LS30 .26
74LS32 .39
74LS38 .39
74LS42 .78
74LS48 .78
74LS51 .25
74LS54 .35
74LS74 .52
74LS75 .65
74LS83 .95
74LS85 1.15
74LS86 .45
74LS90 .70
74LS93 .70
74LS107 .45
74LS112 .48
74LS113 .48
74LS122 .50
74LS123 1.15
74LS126 .86
74LS138 .85
74LS151 .75
74LS153 .75
74LS155 1.15
74LS158 .75
74LS160 .95
74LS161 1.10
74LS162 .95
74LS163 .95
74LS164 1.15
74LS165 .89
74LS170 1.95
74LS174 .90
74LS175 .90
74LS190 1.10
74LS193 .95
74LS195 .95
74LS196 .85
74LS221 1.40
74LS240 2.45
74LS241 2.45
74LS243 2.20
74LS244 2.45
74LS245 6.95
74LS253 .95
74LS257 .95
74LS258 .95
74LS259 2.85
74LS279 .55
74LS283 1.00
74LS293 1.85
74LS298 1.20
74LS366 .95
74LS367 .95
74LS368 .95
74LS373 2.50
74LS374 2.50
74LS386 .65
SN74181N 1.55
SN74393N 1.75
APPLE II Computer with full 48K of memory!
$1099.00
APPLE EXPANSION KIT
16K Memory Add-On
$47.50
MEMORY ADD-ON KIT INCLUDES INSTRUCTIONS, RAMS, AND JUMPERS. NO TOOLS REQUIRED
Plexiglass cover as shown
$24.95
SOROC TECHNOLOGY, INC.
IQ120 $699.00
PROTO CLIPS
14 PIN . . . $4.35
16 PIN . . . $4.50
24 PIN . . . $8.75
40 PIN . . . $14.75
video 100
$139.00
Leedex Corp.
12" BLACK & WHITE LOW COST VIDEO MONITOR
80-COLUMN IMPACT PRINTER
LOWEST IN PRICE—HIGHEST IN PERFORMANCE
*Compact Size
*60 Lines per Minute
*115/230VAC, 50 or 60 Hz.
*72, 80, 96, 120 or 132 Characters/Line
*Self-test Switch
*Auxiliary User Defined Character Set
*RS-232, 20ma, IEEE 488 and Centronics I/O
*16 Baud Rates to 19,200
$649.00
base2, inc.
8038C VCO Waveform Gen. w/sine $2.65
2114L 1024×4 Static RAM 450 ns $4.50
DUST COVERS
APPLE DISK $ 3.75
APPLE KEYBOARD $ 8.75
TRS-80 KEYBOARD $ 7.75
TRS-80 RECORDER $ 3.75
TRS-80 SINGLE DISK $ 3.75
TRS-80 DOUBLE DISK $ 6.75
PET DISK $ 7.75
PET TERMINAL/COMP. $15.75
NORTHSTAR COMPUTER $11.75
SOROC TERMINAL $10.75
Each course below includes a special course book plus two cassettes, for a total course length of 2½ hours. The lecture is completely coordinated to the pages of the book, and cassettes can be played on any standard cassette player.
HOME STUDY COURSE ON CASSETTE
S1·INTRODUCTION TO MICROPROCESSORS
This seminar is intended for all non-specialists who wish to acquire a broad understanding of the basic concepts and advantages of microprocessors. It explains how microprocessors work, and it stresses methods, costs, advantages and disadvantages for the most important application areas of each type of microprocessor. What is needed to implement a system; how to use it; the impact of microprocessor-based systems; their evolution. Topics covered include: BASIC DEFINITIONS, SYSTEM COMPONENTS, MICROPROCESSOR APPLICATIONS, WHAT TO LOOK FOR, and IMPACT AND EVALUATION.
$29.95
S2·PROGRAMMING MICROPROCESSORS
This seminar describes the internal operation of a microprocessor system including how instructions are fetched and executed, how programs are written and executed in typical cases (arithmetic and input/output). The goal of this course is to provide an overall understanding of the concepts of microprocessor programming. Requires an understanding of the main concepts in the INTRODUCTION TO MICROPROCESSORS SEMINAR. It is recommended that these two seminars be taken together.
$29.95
·NO TECHNICAL BACKGROUND ASSUMED·
POWER CONTROLLER
8 outlet · 6 switched
EMI filtered
Circuit Breaker
$87.50
EPROM'S
2708 $6.75
1K×8 450NS
8 FOR $48.50
2716 $18.95
16K(2K×8) 450NS
8 FOR $142.95
2732 $99.50 $58.50
32K(4096×8)
$17.95
LOGIC PROBE KIT
TRS-80
16K Memory Add-On
$49.95 KIT
With jumpers and instructions
CONCORD COMPUTER COMPONENTS
1971 SOUTH STATE COLLEGE · ANAHEIM, CA. 92806
(714) 937-0637
VISA · MASTER CHARGE
CHECK OR M.O.
NO C.O.D.
We stock and sell over 12,000 types of semi-conductors
MINIMUM ORDER $10.00
ADD $1.50 FOR FRT.
CAL. RES. ADD 6%
VOLTAGE REGULATORS
NEGATIVE
7905 (5V)
7908 (8V)
7915 (15V)
7918 (18V)
POSITIVE
7805 (5V)
7806 (6V)
7808 (8V)
7812 (12V)
7815 (15V)
7818 (18V)
7824 (24V)
95¢

7400 TTL

		SN7470N	.29		
		SN7472N	.29		
SN7400N	.20	SN7473N	.35	SN74160N	.89
SN7401N	.20	SN7474N	.35	SN74161N	.89
SN7402N	.20	SN7475N	.49	SN74162N	1.95
SN7403N	.20	SN7476N	.35	SN74163N	.89
SN7404N	.25	SN7479N	5.00	SN74164N	.89
SN7405N	.20	SN7480N	.50	SN74165N	.89
SN7406N	.29	SN7482N	.99	SN74166N	1.25
SN7407N	.29	SN7483N	.69	SN74167N	1.95
SN7408N	.20	SN7485N	.89	SN74170N	1.59
SN7409N	.20	SN7486N	.35	SN74172N	6.00
SN7410N	.18	SN7489N	1.75	SN74173N	1.25
SN7411N	.25	SN7490N	.45	SN74174N	1.00
SN7412N	.25	SN7491N	.59	SN74175N	1.00
SN7413N	.40	SN7492N	.43	SN74176N	.79
SN7414N	.70	SN7493N	.43	SN74177N	.79
SN7416N	.25	SN7494N	.65	SN74179N	1.95
SN7417N	.25	SN7495N	.65	SN74180N	.79
SN7420N	.20	SN7496N	.65	SN74181N	1.95
SN7421N	.29	SN7497N	3.00	SN74182N	.79
SN7422N	.39	SN74100N	1.25	SN74184N	1.95
SN7423N	.25	SN74107N	.35	SN74185N	1.95
SN7425N	.29	SN74109N	.59	SN74186N	9.95
SN7426N	.29	SN74116N	1.95	SN74188N	3.95
SN7427N	.25	SN74121N	.35	SN74190N	1.25
SN7429N	.39	SN74122N	.39	SN74191N	1.25
SN7430N	.20	SN74123N	.59	SN74192N	.79
SN7432N	.25	SN74125N	.49	SN74193N	.79
SN7437N	.25	SN74126N	.49	SN74194N	.89
SN7438N	.40	SN74132N	.75	SN74195N	.69
SN7439N	.25	SN74136N	.75	SN74196N	.89
SN7440N	.20	SN74141N	.79	SN74197N	.89
SN7441N	.89	SN74142N	2.95	SN74198N	1.49
SN7442N	.59	SN74143N	2.95	SN74199N	1.49
SN7443N	.75	SN74144N	2.95	SN74S200	4.95
SN7444N	.75	SN74145N	.79	SN74251N	.99
SN7445N	.75	SN74147N	1.95	SN74279N	.79
SN7446N	.69	SN74148N	1.29	SN74283N	2.25
SN7447N	.59	SN74150N	1.25	SN74284N	3.95
SN7448N	.79	SN74151N	.59	SN74285N	3.95
SN7450N	.20	SN74152N	.59	SN74365N	.69
SN7451N	.20	SN74153N	.59	SN74366N	.69
SN7453N	.20	SN74154N	1.50	SN74367N	.69
SN7454N	.20	SN74155N	.79	SN74368N	.69
SN7459A	.25	SN74156N	.79	SN74390N	1.95
SN7460N	.20	SN74157N	.65	SN74393N	1.95

CMOS

CD4000	.39			CD4070	.55
CD4001	.39			CD4071	.49
CD4002	.39	CD4028	.89	CD4072	.49
CD4006	1.19	CD4029	1.49	CD4076	1.39
CD4007	.25	CD4030	.49	CD4081	.39
CD4009	.49	CD4035	.99	CD4082	.39
CD4010	.49	CD4040	1.49	CD4093	.99
CD4011	.39	CD4041	1.49	CD4098	1.19
CD4012	.25	CD4042	.99	MC14409	14.95
CD4013	.49	CD4043	.89	MC14410	14.95
CD4014	1.39	CD4044	.89	MC14411	14.95
CD4015	1.19	CD4046	1.79	MC14419	4.95
CD4016	.59	CD4047	2.50	MC14433	13.95
CD4017	1.19	CD4048	1.35	MC14506	.75
CD4018	.99	CD4049	.49	MC14507	.99
CD4019	.49	CD4050	.69	MC14562	11.95
CD4020	1.19	CD4051	1.19	MC14583	2.49
CD4021	1.39	CD4053	1.19	CD4508	3.95
CD4022	1.19	CD4056	2.95	CD4510	1.39
CD4023	.29	CD4059	9.95	CD4511	1.29
CD4024	.79	CD4060	1.49	CD4515	2.95
CD4025	.23	CD4066	.79	CD4518	1.29
CD4026	2.95	CD4068	.39	CD4520	1.29
CD4027	.69	CD4069	.45	CD4566	2.25

74C00

74C00	.39			74C163	2.49
74C02	.39			74C164	2.49
74C04	.45	74C85	2.49	74C173	2.60
74C08	.49	74C90	1.95	74C192	2.49
74C10	.39	74C93	1.95	74C193	2.49
74C14	1.95	74C95	1.95	74C195	2.49
74C20	.39	74C107	1.25	74C922	7.95
74C30	.39	74C151	2.90	74C923	6.25
74C42	1.95	74C154	3.00	74C925	8.95
74C48	2.49	74C157	2.15	74C926	8.95
74C73	.89	74C160	2.49	80C95	1.50
74C74	.89	74C161	2.49	80C97	1.50

LINEAR

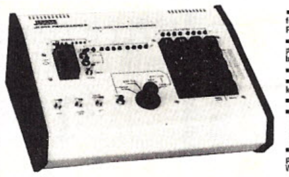
78MG	1.75			LM710N	.79
LM106H	.99			LM711N	.39
LM300H	.80			LM723N/H	.55
LM301CN/H	.35	LM340K-18	1.35	LM733N	1.00
LM302H	.75	LM340K-24	1.35	LM739N	1.19
LM304H	1.00	LM340T-5	1.25	LM741CN/H	.35
LM305H	.60	LM340T-6	1.25	LM741-14N	.39
LM307CN/H	.35	LM340T-8	1.25	LM747N/H	.79
LM308CN/H	1.00	LM340T-12	1.25	LM748N/H	.39
LM309H	1.10	LM340T-15	1.25	LM1310N	1.95
LM309K	1.25	LM340T-18	1.25	LM1458CN/H	.59
LM310CN	1.95	LM340T-24	1.25	MC1488N	1.95
LM311N/H	.90	LM358N	1.00	MC1489N	1.95
LM312H	1.95	LM370N	1.95	LM1496N	.95
LM317K	6.50	LM373N	3.25	LM1556V	1.75
LM318CN/H	1.50	LM377N	4.00	MC1741SCP	3.00
LM319N	1.30	LM380N	1.25	LM2111N	1.95
LM320K-5	1.35	LM380CN	.99	LM2901N	2.95
LM320K-5.2	1.35	LM381N	1.79	LM3053N	1.50
LM320K-12	1.25	LM382N	1.79	LM3065N	1.49
LM320K-15	1.35	NE501N	8.00	LM3900N(3401)	.59
LM320K-18	1.35	NE510A	6.00	LM3905N	1.49
LM320K-24	1.35	NE529A	4.95	LM3909N	1.25
LM320T-5	1.25	NE531H/V	3.95	MC5558V	.59
LM320T-5.2	1.25	NE536T	6.00	8038B	4.95
LM320T-8	1.25	NE540	6.00	LM75450N	.49
LM320T-12	1.25	NE544N	4.95	75451CN	.39
LM320T-15	1.25	NE550N	1.30	75452CN	.39
LM320T-18	1.25	NE555V	.39	75453CN	.39
LM320T-24	1.25	NE556N	.99	75454CN	.39
LM323K-5	5.95	NE560B	5.00	75491CN	.79
LM324N	1.49	NE562B	5.00	75492CN	.89
LM339N	.99	NE565N/H	1.25	75493N	.89
LM340K-5	1.35	NE566CN	1.75	75494CN	.89
LM340K-6	1.35	NE567V/H	.99	RC4136	1.25
LM340K-8	1.35	NE570N	4.95	RC4151	3.95
LM340K-12	1.35	LM703CN/H	.69	RC4194	4.95
LM340K-15	1.35	LM709N/H	.29	RC4195	4.49

74LS00TTL

74LS00	.35			74LS139	1.05
74LS01	.35	74LS51	.29	74LS151	1.05
74LS02	.35	74LS54	.29	74LS155	1.05
74LS03	.35	74LS55	.29	74LS157	1.05
74LS04	.42	74LS73	.54	74LS160	1.15
74LS05	.42	74LS74	.54	74LS161	1.39
74LS08	.35	74LS75	.71	74LS162	1.25
74LS09	.42	74LS76	.54	74LS163	1.39
74LS10	.35	74LS78	.49	74LS164	1.50
74LS11	.75	74LS83	1.05	74LS175	1.25
74LS13	.59	74LS85	1.50	74LS181	2.49
74LS14	1.25	74LS86	.54	74LS190	1.39
74LS15	.35	74LS90	.71	74LS191	1.39
74LS20	.35	74LS92	.90	74LS192	1.39
74LS21	.35	74LS93	.90	74LS193	1.39
74LS22	.35	74LS95	.99	74LS194	1.39
74LS26	.35	74LS96	1.15	74LS195	1.39
74LS27	.35	74LS107	.54	74LS253	1.25
74LS28	.35	74LS109	.54	74LS257	1.05
74LS30	.35	74LS112	.54	74LS258	1.75
74LS32	.42	74LS123	1.50	74LS260	.83
74LS37	.45	74LS125	1.05	74LS279	.90
74LS40	.35	74LS132	.99	74LS367	.75
74LS42	1.05	74LS136	.59	74LS368	.75
74LS47	1.05	74LS138	1.05	74LS670	2.49

JE608 PROGRAMMER

2704/2708 EPROM PROGRAMMER



- 3 separate Display Registers: 8 LED's for Hex Key entries, 10 LED's (2^0-2^9) for Address Register and 8 LED's for Data Memory Register. The Data Memory Register displays the content of the RAMs from the EPROM Chip.
- Development of microprocessor systems by means of a ribbon cable from the programmer panel test socket to the EPROM socket on the microprocessor board.
- Rapid checking verification of programmed data changes.
- User may move data from a master to RAM's or write into RAM's with keyboard entries.
- Allows manual stepping manipulation (up and down) at any address location.
- Stand-alone EPROM Programmer consisting of: A 19-key Hexadecimal Keyboard assembly, Programmer Board assembly with 4 power supplies and a LED/Test Socket Panel Board assembly. The Test Socket is zero force insertion type. Power requirements: 115VAC, 60Hz, 6W.
- Compact desk-top enclosure: Color-coordinated designer's case with light tan panels and molded end pieces in mocha brown. Size: 3¾"H x 11"W x 8¾"D. Weight: 5 lbs.

The JE608 EPROM Programmer is a completely self-contained unit which is independent of computer control and requires no additional systems for its operations. The EPROM can be programmed from the Hexadecimal Keyboard or from a pre-programmed EPROM. The JE608 Programmer can emulate a programmed EPROM by the use of its internal RAM circuits. This will allow the user to test or pretest a program, for a system, prior to programming a chip. Any changes in the program can be entered directly into the memory circuits with the Hexadecimal Keyboard so that rewriting the entire program will not be necessary. The JE608 Programmer contains a Programmer/Board with 25 IC's and including power supplies of: −5V, +5V, +12V and +26V. The Hexadecimal Keyboard and LED/Test Socket Panel Board are separate assemblies within the system.

JE608 KIT $399.95
JE608 Assembled and Tested $499.95

DISCRETE LEDS

XC556R	.200"	red	5/$1	MV50	.085"	red	6/$1	XC111R	.190"	red	5/$1
XC556G	.200"	green	4/$1	XC209R	.125"	red	5/$1	XC111G	.190"	green	4/$1
XC556Y	.200"	yellow	4/$1	XC209G	.125"	green	4/$1	XC111Y	.190"	yellow	4/$1
XC556C	.200"	clear	4/$1	XC209Y	.125"	yellow	4/$1	XC111C	.190"	clear	4/$1
XC22R	.200"	red	5/$1	XC526R	.185"	red	5/$1				
XC22G	.200"	green	4/$1	XC526G	.185"	green	4/$1				
XC22Y	.200"	yellow	4/$1	XC526Y	.185"	yellow	4/$1	INFRA-RED LED			
MV10B	.170"	red	4/$1	XC526C	.185"	clear	4/$1	¼"x¼"x1/16"	flat		5/$1

DISPLAY LEDS

TYPE	POLARITY	HT	PRICE	TYPE	POLARITY	HT	PRICE
MAN 1	Common Anode-red	.270	2.95	MAN 6730	Common Anode-red ± 1	.560	.99
MAN 2	5 x 7 Dot Matrix-red	.300	4.95	MAN 6740	Common Cathode-red-D.D.	.560	.99
MAN 3	Common Cathode-red	.125	.25	MAN 6750	Common Cathode-red ± 1	.560	.99
MAN 4	Common Cathode-red	.187	1.95	MAN 6760	Common Anode-red	.560	.99
MAN 7G	Common Anode-green	.300	.25	MAN 6780	Common Cathode-red	.560	.99
MAN 7Y	Common Anode-yellow	.300	.99	DL701	Common Anode-red ± 1	.300	.99
MAN 72	Common Anode-red	.300	.75	DL704	Common Cathode-red	.300	.99
MAN 74	Common Cathode-red	.300	1.25	DL707	Common Anode-red	.300	.99
MAN 82	Common Anode-yellow	.300	.49	DL728	Common Cathode-red	.500	1.49
MAN 84	Common Cathode-yellow	.300	.99	DL741	Common Anode-red	.600	1.25
MAN 3620	Common Anode-orange	.300	.49	DL746	Common Anode-red ± 1	.630	1.49
MAN 3630	Common Anode-orange ± 1	.300	.99	DL747	Common Anode-red	.600	1.49
MAN 3640	Common Cathode-orange	.300	.99	DL749	Common Cathode-red ± 1	.630	1.49
MAN 4610	Common Anode-orange	.300	.99	DL750	Common Cathode-red	.600	1.49
MAN 4640	Common Cathode-orange	.400	.99	DL338	Common Cathode-red	.110	.35
MAN 4710	Common Anode-red	.400	.99	FND70	Common Cathode	.250	.69
MAN 4730	Common Anode-red ± 1	.400	.99	FND358	Common Cathode ± 1	.357	.99
MAN 4740	Common Cathode-red	.400	.99	FND359	Common Cathode	.357	.75
MAN 4810	Common Anode-yellow	.400	.99	FND503	Common Cathode(FND500)	.500	.99
MAN 4840	Common Cathode-yellow	.400	.99	FND507	Common Anode (FND510)	.500	.99
MAN 6610	Common Anode-orange-D.D.	.560	.99	5082-7730	Common Anode-red	.300	.99
MAN 6630	Common Anode-orange ± 1	.560	.99	HDSP-3400	Common Anode-red	.800	1.50
MAN 6640	Common Cathode-orange-D.D.	.560	.99	HDSP-3403	Common Cathode red	.800	1.50
MAN 6650	Common Cathode-orange ± 1	.560	.99	5082-7300	4 x 7 sgl. Digit-RHDP	.600	19.95
MAN 6660	Common Anode-orange	.560	.99	5082-7302	4 x 7 Sgl. Digit-LHDP	.600	19.95
MAN 6680	Common Cathode-orange	.560	.99	5082-7304	Overrange character (±1)	.600	15.00
MAN 6710	Common Anode-red-D.D.	.560	.99	5082-7340	4 x 7 Sgl. Digit-Hexadecimal	.600	22.50

RCA LINEAR

CA3013T	2.15	CA3082N	2.00
CA2023T	3.25	CA3083N	1.60
CA3035T	2.48	CA3086N	.85
CA3039T	1.35	CA3089N	3.75
CA3046N	1.30	CA3130T	1.39
CA3059N	3.25	CA3140T	1.25
CA3060N	3.25	CA3160T	1.25
CA3080T	1.25	CA3401N	.59
CA3081N	2.00	CA3600N	3.50

CALCULATOR CHIPS/DRIVERS

MM5725	$2.95
MM5738	2.95
DM8864	2.00
DM8865	1.00
DM8887	.75
DM8889	.75
9374 7-seg. LED driver	1.50
MM5309	4.95

CLOCK CHIPS

MM5309	4.95
MM5311	4.95
MM5312	4.95
MM5314	4.95
MM5316	6.95
MM5318	9.95
MM5369	2.95
MM5387/1998a	4.95
CT 7001	6.95

MOTOROLA

MC1408L7	4.95
MC1408L8	5.75
MC1439L	2.95
MC3022P	2.95
MC3061P	3.50
MC4016(74416)	7.50
MC4024P	3.95
MC4040P	6.95
MC4044P	4.50

LOW PROFILE (TIN) SOCKETS

	1-24	25-49	50-100
8 pin LP	.17	.16	.15
14 pin LP	.20	.19	.18
16 pin LP	.22	.21	.20
18 pin LP	.29	.28	.27
20 pin LP	.34	.32	.30
22 pin LP	.37	.36	.35
24 pin LP	.38	.37	.36
28 pin LP	.45	.44	.43
36 pin LP	.60	.59	.58
40 pin LP	.63	.62	.61

SOLDERTAIL STANDARD (TIN)

	1-24	25-49	50-100
14 pin ST	.27	.25	.24
16 pin ST	.30	.27	.25
18 pin ST	.35	.32	.30
24 pin ST	.49	.45	.42
28 pin ST	.99	.90	.81
36 pin ST	1.39	1.26	1.15
40 pin ST	1.59	1.45	1.30

SOLDERTAIL (GOLD) STANDARD

	1-24	25-49	50-100
8 pin SG	.39	.35	.31
14 pin SG	.49	.45	.41
16 pin SG	.54	.49	.44
18 pin SG	.59	.53	.48
24 pin SG	.79	.75	.69
28 pin SG	1.10	1.00	.90
36 pin SG	1.65	1.40	1.26
40 pin SG	1.75	1.59	1.45

WIRE WRAP SOCKETS (GOLD) LEVEL #3

	1-24	25-49	50-100
8 pin WW	.59	.54	.49
10 pin WW	.69	.63	.58
14 pin WW	.79	.73	.67
16 pin WW	.85	.77	.70
18 pin WW	.99	.90	.81
20 pin WW	1.19	1.08	.99
22 pin WW	1.49	1.35	1.23
24 pin WW	1.39	1.26	1.14
28 pin WW	1.69	1.53	1.38
36 pin WW	2.19	1.99	1.79
40 pin WW	2.29	2.09	1.89

1/4 WATT RESISTOR ASSORTMENTS – 5%

ASST. 1	5 ea.	10 OHM / 27 OHM	12 OHM / 33 OHM	15 OHM / 39 OHM	18 OHM / 47 OHM	22 OHM / 56 OHM	50 PCS	$1.75
ASST. 2	5 ea.	68 OHM / 180 OHM	82 OHM / 220 OHM	100 OHM / 270 OHM	120 OHM / 330 OHM	150 OHM / 390 OHM	50 PCS	1.75
ASST. 3	5 ea.	470 OHM / 1.2K	560 OHM / 1.5K	680 OHM / 1.8K	820 OHM / 2.2K	1K / 2.7K	50 PCS	1.75
ASST. 4	5 ea.	3.3K / 8.2K	3.9K / 10K	4.7K / 12K	5.6K / 15K	6.8K / 18K	50 PCS	1.75
ASST. 5	5 ea.	22K / 56K	27K / 68K	33K / 82K	39K / 100K	47K / 120K	50 PCS	1.75
ASST. 6	5 ea.	150K / 390K	180K / 470K	220K / 560K	270K / 680K	330K / 820K	50 PCS	1.75
ASST. 7	5 ea.	1M / 2.7M	1.2M / 3.3M	1.5M / 3.9M	1.8M / 4.7M	2.2M / 5.6M	50 PCS	1.75

ASST. 8R Includes Resistor Assortments 1-7 (350 PCS.) **$9.95 ea.**

$10.00 Min. Order — U.S. Funds Only
Calif. Residents Add 6% Sales Tax
Postage — Add 5% plus $1 Insurance (if desired)

Spec Sheets — 25¢
1980 Catalog Available — Send 41¢ stamp

FREE 1980 CATALOG

Jameco ELECTRONICS

PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS — WORLDWIDE

1355 SHOREWAY ROAD, BELMONT, CA 94002

7-80 PRICES SUBJECT TO CHANGE

TELEPHONE/KEYBOARD CHIPS

AY-5-9100	Push Button Telephone Dialler	$14.95
AY-5-9200	Repertory Dialler	14.95
AY-5-9500	CMOS Clock Generator	4.95
AY-5-2376	Keyboard Encoder (88 keys)	14.95
HD0165	Keyboard Encoder (16 keys)	7.95
74C922	Keyboard Encoder (16 keys)	7.95
74C923	Keyboard Encoder (20 keys)	6.25

ICM CHIPS

ICM7045	CMOS Precision Timer	24.95
ICM7205	CMOS LED Stopwatch/Timer	19.95
ICM7207	Oscillator Controller	7.50
ICM7208	Seven Decade Counter	19.95
ICM7209	Clock Generator	6.95

NMOS READ ONLY MEMORIES

MCM6571	128 X 9 X 7 ASCII Shifted with Greek	13.50
MCM6574	128 X 9 X 7 Math Symbol & Pictures	13.50
MCM6575	128 X 9 X 7 Alpha Control Char. Gen	13.50

MISCELLANEOUS

TL074CN	Quad Low Noise bi-fet Op Amp	2.49
TL494CN	Switching Regulator	4.49
TL496CP	Single Switching Regulator	1.75
11C90	Divide 10/11 Prescaler	19.95
95H90	Hi-Speed Divide 10/11 Prescaler	11.95
4N33	Photo-Darlington Opto-Isolator	3.95
MK50240	Top Octave Freq. Generator	17.50
DS0026CH	5Mhz 2-phase MOS clock driver	3.75
TIL308	.27" red num. display w/integ. logic chip	10.95
MM5320	TV Camera Sync. Generator	14.95
MM5330	4½ Digit DPM Logic Block **(Special)**	3.95
LD110/111	3½ Digit A/D Converter Set	25.00/set
MC14433P	3½ Digit A/D Converter	13. 95

LITRONIX ISO-LIT 1
Photo Transistor Opto-Isolator
(Same as MCT 2 or 4N25)
49¢ each

SN 76477
SOUND GENERATOR
Generates Complex Sounds
Low Power - Programmable
$3.95 each

TV GAME CHIP AND CRYSTAL
AY-3-8500-1 and 2.01 MHZ Crystal (Chip & Crystal includes score display, 6 games and select angles, etc. **7.95/set**

EXAR

XR205	$8.40			XR2242CP	1.50
XR210	4.40			XR2264	4.25
XR215	4.40			XR2556	3.20
XR320	1.55			XR2567	2.99
XR-L555	1.50	JE2206KB	19.95	XR3403	1.25
XR555	.39	XR1800	3.20	XR4136	1.25
XR556	.99	XR2206	4.40	XR4151	3.95
XR567CP	.99	XR2207	3.85	XR4194	1.45
XR567CT	1.25	XR2208	5.20	XR4202	3.60
XR1310P	1.95	XR2209	1.75	XR4212	2.05
XR1468CN	3.85	XR2211	5.25	XR4558	.75
XR1488	1.95	XR2212	4.35	XR4739	1.15
XR1489	1.95	XR2240	3.45	XR4741	1.47

DIODES

TYPE	VOLTS	W	PRICE	TYPE	VOLTS	W	PRICE
				1N4002	100 PIV	1 AMP	12/1.00
				1N4003	200 PIV	1 AMP	12/1.00
1N746	3.3	400m	4/1.00	1N4004	400 PIV	1 AMP	12/1.00
1N751	5.1	400m	4/1.00	1N4005	600 PIV	1 AMP	10/1.00
1N752	5.6	400m	4/1.00	1N4006	800 PIV	1 AMP	10/1.00
1N753	6.2	400m	4/1.00	1N4007	1000 PIV	1 AMP	10/1.00
1N754	6.8	400m	4/1.00	1N3600	50	200m	6/1.00
1N757	9.0	400m	4/1.00	1N4148	75	10m	15/1.00
1N759	12.0	400m	4/1.00	1N4154	35	10m	12/1.00
1N959	8.2	400m	4/1.00	1N4733	5.1	1w	28
1N965	15	400m	4/1.00	1N4734	5.6	1w	28
1N5232	5.6	500m	28	1N4735	6.2	1w	28
1N5234	6.2	500m	28	1N4736	6.8	1w	28
1N5235	6.8	500m	28	1N4738	8.2	1w	28
1N5236	7.5	500m	28	1N4742	12	1w	28
1N5242	12	500m	28	1N4744	15	1w	28
1N5245	15	500m	28	1N1183	50 PIV	35 AMP	1.60
1N456	25	40m	6/1.00	1N1184	100 PIV	35 AMP	1.70
1N458	150	7m	6/1.00	1N1185	150 PIV	35 AMP	1.70
1N485A	180	10m	5/1.00	1N1186	200 PIV	35 AMP	1.80
1N4001	50 PIV	1 AMP	12/1.00	1N1188	400 PIV	35 AMP	3.00

SCR AND FW BRIDGE RECTIFIERS

C36D	15A @ 400V	SCR(2N1849)	$1.95
C36M	35A @ 600V	SCR	1.95
2N2328	1.6A @ 300V	SCR	.50
MDA 980-1	12A @ 50V	FW BRIDGE REC.	1.95
MDA 980-3	12A @ 200V	FW BRIDGE REC.	1.95

TRANSISTORS

C106B1	.50			2N3904	4/1.00
MPSA05	.30	2N3055	.89	2N3905	4/1.00
MPSA06	5/1.00	MJE3055	1.00	2N3906	4/1.00
TIS97	6/1.00	2N3392	5/1.00	2N4013	3/1.00
TIS98	6/1.00	2N3398	5/1.00	2N4123	6/1.00
40409	1.75	PN3567	3/1.00	PN4249	4/1.00
40410	1.75	PN3568	4/1.00	PN4250	4/1.00
40673	1.75	PN3569	4/1.00	2N4400	4/1.00
2N918	4/1.00	MPS3638A	5/1.00	2N4401	4/1.00
2N2219A	2/1.00	MPS3702	5/1.00	2N4402	4/1.00
2N2221A	4/1.00	2N3704	5/1.00	2N4403	4/1.00
2N2222A	5/1.00	MPS3704	5/1.00	2N4409	5/1.00
PN2222 Plastic	7/1.00	2N3705	5/1.00	2N5086	4/1.00
2N2369A	4/1.00	MPS3705	5/1.00	2N5087	4/1.00
MPS2369	5/1.00	2N3706	5/1.00	2N5088	4/1.00
2N2484	4/1.00	MPS3706	5/1.00	2N5089	4/1.00
2N2906	4/1.00	2N3707	5/1.00	2N5129	5/1.00
2N2907	5/1.00	2N3711	5/1.00	PN5134	5/1.00
		2N3724A	.65	PN5138	5/1.00
2N2925	5/1.00	2N3725A	1.00	2N5139	5/1.00
MJE2955	1.25	2N3772	2.25	2N5210	5/1.00
2N3053	2/1.00	2N3823	1.00	2N5449	3/1.00
		2N3903	5/1.00	2N5951	3/1.00

CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

	1-9	10-99	100+		1-9	10-99	100+
10 pf	.05	.04	.03	.001µF	.05	.04	.035
22 pf	.05	.04	.03	.0047µF	.05	.04	.035
47 pf	.05	.04	.03	.01µF	.05	.04	.035
100 pf	.05	.04	.03	.022µF	.06	.05	.04
220 pf	.05	.04	.03	.047µF	.06	.05	.04
470 pf	.05	.04	.035	.1µF	.12	.09	.075

100 VOLT MYLAR FILM CAPACITORS

.001mf	.12	.10	.07	.022mf	.13	.11	.08
.0022	.12	.10	.07	.047mf	.21	.17	.13
.0047mf	.12	.10	.07	.1mf	.27	.23	.17
.01mf	.12	.10	.07	.22mf	.33	.27	.22

+20% DIPPED TANTALUMS (SOLID) CAPACITORS

.1/35V	.39	.31	.25	1.5/35V	.41	.33	.26
.15/35V	.39	.31	.25	2.2/35V	.51	.41	.33
.22/35V	.39	.31	.25	3.3/25V	.53	.43	.34
.33/35V	.39	.31	.25	4.7/25V	.63	.51	.41
.47/35V	.39	.31	.25	6.8/25V	.79	.63	.50
.68/35V	.39	.31	.25	15/25V	1.39	1.12	.89
1.0/35V	.39	.31	.25	22/6V	.79	.63	.50

MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS

Axial Lead				Radial Lead			
.47/50V	.15	.13	.10	.47/25V	.15	.13	.10
1.0/50V	.16	.14	.11	.47/50V	.16	.14	.11
3.3/50V	.14	.12	.09	1.0/16V	.15	.13	.10
4.7/25V	.15	.13	.10	1.0/25V	.16	.14	.11
10/25V	.15	.13	.10	1.0/50V	.16	.14	.11
10/50V	.16	.14	.12	4.7/16V	.15	.13	.10
22/25V	.17	.15	.12	4.7/25V	.15	.13	.10
22/50V	.24	.20	.18	4.7/50V	.16	.14	.11
47/25V	.19	.17	.15	10/16V	.14	.12	.09
47/50V	.25	.21	.19	10/25V	.15	.13	.10
100/25V	.24	.20	.18	10/50V	.16	.14	.12
100/50V	.35	.30	.28	47/50V	.24	.21	.19
220/25V	.32	.28	.25	100/16V	.19	.15	.14
220/50V	.45	.41	.38	100/25V	.24	.20	.18
470/25V	.33	.29	.27	100/50V	.35	.30	.28
1000/16V	.55	.50	.45	220/16V	.23	.17	.16
2200/16V	.70	.62	.55	470/25V	.31	.28	.26

WAMECO

THE COMPLETE PC BOARD HOUSE
EVERYTHING FOR THE S-100 BUSS

* **FPB-1** FRONT PANEL BOARD FOR 8080A AND Z80 SYSTEMS IMSAI COMPATIBLE.
  PCBD .... $54.95 KIT $165.00
* **MEM-2** 16K RAM 2114's. ADDRESSABLE IN 4K BOUNDARIES.
  PCBD .... $31.95 KIT (LESS RAMS) $80.95
* **EPM-2** 16/32K ROM USES 2716 OR 2708. ADDRESSABLE IN 4K BOUNDARIES.
  PCBD .... $31.95 KIT (LESS ROMS) $74.95
* **CPU-1** 8080A PROCESSOR BOARD WITH VECTOR INTERRUPT.
  PCBD .... $31.95 KIT $124.95
* **IOB-1** I/O BOARD. ONE SERIAL, TWO PARALLEL WITH CASSETTE. PCBD $31.95
* **FDC-1** FLOPPY DISC CONTROLLER BOARD USES 1771. PCBD $44.95
* **QMB-12** 13 SLOT MOTHER BOARD.
  PCBD .... $39.95 KIT $115.95
* **QMB-9** 9 SLOT MOTHER BOARD.
  PCBD .... $34.95 KIT $89.95
* **PTB-1** POWER SUPPLY AND TERMINATOR BOARD.
  PCBD .... $29.95 KIT $49.95
* **RTC-1** REAL TIME CLOCK BOARD WITH TWO INTERRUPTS.
  PCBD .... $27.95 KIT $79.95
* **MEM-1** 8K RAM, USES 2102's.
  PCBD .... $31.95 KIT (LESS RAM) $71.95
* **EPM-1** 4K 170Z BOARD.
  PCBD .... $29.95 KIT (LESS ROM) $59.95

FUTURE PRODUCTS: 80 CHARACTER VIDEO BOARD.
Z-80 CPU BOARD WITH ROM, 8 PARALLEL PORT I/O BOARD.

DEALER INQUIRIES INVITED, UNIVERSITY DISCOUNTS AVAILABLE
AT YOUR LOCAL DEALER

MOST PRODUCTS FOR IMMEDIATE SHIPMENT. NO 4-8 WEEK DELAYS REQUIRED FOR OTHERS.

WMC inc.

WAMECO, INC., P. O. BOX 877 • 455 PLAZA ALHAMBRA • EL GRANADA, CA 94018 • (415) 726-6378

CALIFORNIA COMPUTER SYSTEMS

16K RAM BOARD. Fully buffered addressable in 4K blocks. IEEE standard for bank addressing 2114's. PCBD ....$27.95 Kit 450 NSEC $249.95

PT-1 PROTO BOARD. Over 2,600 holes 4" regulators. All S-100 buss functions labeled, gold fingers. PCBD $26.95

PT-2 PROTO BOARD. Similar to PT-1 except set-up to handle solder tail sockets. PCBD ... $26.95

CCS MAIN FRAME. Kit (S-100) $339.95

APPLE EXTENDER. Kit $22.95

APPLE IEEE INSTRUMENTATION INTERFACE KIT 7490. Kit $275.00

ARITHMETIC PROCESSOR FOR APPLE 7811A. Kit $350.00

APPLE ASYNCHRONOUS SERIAL INTERFACE 7710A. Kit $89.95

APPLE SYNCHRONOUS SERIAL INTERFACE 7712A. Kit $89.95

ALL OTHER CCS PRODUCTS AVAILABLE

SSM

PB-1 2708 & 2716 Programming Board with provisions for 4K or 8K EPROM. No external supplies required. Textool sockets. Kit $129.95

CB-1A 8080 Processor Board. 2K of PROM 256 BYTE RAM power on/rest Vector Jump Parallel port with status. Kit ....$129.95 PCBD $27.95

VB-3 80 x 55 VIDEO BOARD. Graphics included. 2 MHZ .... $294.95 4 MHZ $329.95

IO-4 Two serial I/O ports with full handshaking 20/60 ma current loop: Two parallel I/O ports. Kit ....$130.00 PCBD $27.95

VB-1B 64 x 16 video board, upper lower case Greek composite and parallel video with software, S-100. Kit ....$125.00 PCBD $27.95

CB-2 Z80 CPU BOARD. Kit $185.95

AIO APPLE SERIAL/PARALLEL $125.95

ALL OTHER SSM PRODUCTS AVAILABLE

WMC inc. WAMECO INC.

FDC-1 FLOPPY CONTROLLER BOARD will drive shugart, pertek, remic 5" & 8" drives up to 8 drives, on board PROM with power boot up, will operate with CPM™ (not included). PCBD $42.95

FPB-1 Front Panel. IMSAI size, hex displays. Byte, or instruction single step. PCBD $47.50

MEM-1A 8K x 8 fully buffered, S-100, uses 2102 type rams. PCBD $25.95

QM-12 MOTHER BOARD, 13 slot, terminated, S-100 board only $38.95

CPU-1 8080A Processor board S-100 with 8 level vector interrupt. PCBD $27.95

RTC-1 Realtime clock board. Two independent interrupts. Software programmable. PCBD...... $24.95

EPM-1 1702A 4K Eprom card. PCBD $25.95

EPM-2 2708/2716 16K/32K EPROM CARD. PCBD $25.95

QM-9 MOTHER BOARD. Short Version of QM-12. 9 Slots. PCBD $32.95

MEM-2 16K x 8 Fully Buffered 2114 Board. PCBD $27.95

PTB-1 POWER SUPPLY AND TERMINATOR BOARD. PCBD $27.95

IOB-1 SERIAL AND PARALLEL INTERFACE. 2 parallel, one serial and cassette. PCBD $27.95

2708	$ 9.49	2114L 450 NSEC	$5.99
2716	$35.95	2114L 200 NSEC	$6.99

AUG. SPECIAL SALE
ON PREPAID ORDERS
(Charge cards not included on this offer)

WAMECO BARE PCBD SALE. 10% OFF THE PRICE OF WAMECO PCBD WHEN 5 OR MORE ARE PURCHASED.

MIKOS PARTS ASSORTMENT
WITH WAMECO AND CYBERCOM PCBDS

MEM-2 with MIKOS #7 16K ram with L2114 450 NSEC $249.95

MEM-2 with MIKOS #13 16K ram with L2114 250 NSEC $279.95

CPU-1 with MIKOS #2 8080A CPU ....$ 94.95

QM-12 with MIKOS #4 13 slot mother board ....$ 95.95

RTC-1 with MIKOS #5 real time clock........$ 59.95

EMP-1 with MIKOS #10 4K 1702 less EPROMS ....$ 49.95

EPM-2 with MIKOS #11 16-32K EPROMS less EPROMS ....$ 59.95

QM-9 with MIKOS #12 9 slot mother board ....$ 89.95

FPB-1 with MIKOS #14 all parts for front panel $144.95

MIKOS PARTS ASSORTMENTS ARE ALL FACTORY MARKED PARTS. KITS INCLUDE ALL PARTS LISTED AS REQUIRED FOR THE COMPLETE KIT LESS PARTS LISTED. ALL SOCKETS INCLUDED.

LARGE SELECTION OF LS TTL AVAILABLE

MIKOS

(415) 726-7593
P. O. Box 955 • El Granada, CA 94018
Please send for IC, Xistor and Computer parts list

VISA or MASTERCHARGE. Send account number, interbank number, expiration date and sign your order. Approx. postage will be added. Check or money order will be sent post paid in U.S. If you are not a regular customer, please use charge, cashier's check or postal money order. Otherwise there will be a two-week delay for checks to clear. Calif. residents add 6% tax. Money back 30-day guarantee. We cannot accept returned IC's that have been soldered to. Prices subject to change without notice. **$10 minimum order. $1.50 service charge on orders less than $10.00.**

Reader Service index—page 241

SEND $1.00 FOR 52 PAGE CATALOG

PRIORITY ONE ELECTRONICS

CENTRONICS 730 Dot Matrix Printer

SAVE $100.00

PRICE $695.00

STANDARD FEATURES: • 50 Characters/second • Characters/line • 10 characters/inch • 3-way paper handling system • 7 x 7 dot matrix • 96 character ASCII • microprocessor electronics • unidirection print at 5.0 ips • high speed return approximately 10 ips • 21 lpm with 80 columns printed • 58 lpm with 20 columns printed • 80 character buffer • 6 lpi vertical • Centronics Colors and logo

FORMS HANDLING: Roll Paper 8.5 in. x 5.0 dia. with 1 in. core maximum dimension 3.5 in. wide with .38 in. core minimum dimension. Fan Gold 9.0 in. /22.9 cm. wide pin to pin 9.5 in./24.1cm wide overall. Up to 3 ply paper with 2 carbons (total thickness not to exceed .012 inches) Cut Sheet. Maximum width 8.5 inches

RIBBON SYSTEM: Continuous ribbon 9/16" (14mm) wide, 20 yards (18.3 meters) long. Mobius Loop allows printing on upper and lower portion on alternate passes

OPERATOR CONTROLS: Power on/off. Reset Switch -allows disabling of printer without dropping AC

DATA INPUT: 7 or 8 bit ASCII parallel. TTL levels with strobe. Acknowledge pulse indicates that data was received

PHYSICAL DIMENSIONS: Weight less than 10 lbs/5 kg -Width 14.5 inches/37 cm -Depth 11.0 inches/28cm -Height 4.89 inches/13cm -Dimensions exclusive of roll paper holder

SHIPPING WEIGHT 14 lbs.

CEN-730-1 (Parallel Int.)............LIST PRICE: $795.00

TRS-80/APPLE MEMORY EXPANSION KITS

4116's RAMS

100% GUARANTEED — from Leading Manufacturers — 1,000's SETS SOLD

(16Kx1 200/250ns)

8 for $50.00

ADD $3.00 FOR PROGRAMMING JUMPERS FOR TRS-80 KEYBOARD

4116's 100 pcs & UP $5.20 each

1000 pcs & UP $4.45 each

The STAR modem from Livermore

LIST PRICE **199.00**

OUR CORNER THE MARKET PRICE **$139.00**

FEATURE FITS GTE HANDSETS!

The STAR modem from Livermore represents a significant breakthrough in the development of acoustic modems. The small, lightweight case houses a high-performance modem that competes with the highest quality standard-sized couplers available. Yet, because of its cost effective design, the STAR has become the price/performance leader in the industry.

CIRCUITRY

The switchable, four-section bandpass filter provides the user with excellent out-of-band rejection to assure accurate processing of the received carrier, even at signal levels of less than −47 dBm. Further, the proven soft limiter and phase lock loop discriminator yields data that is essentially jitter free.

The oscillator is built using highly stable, state-variable circuitry that delivers a nearly harmonic free, phase coherent sine wave to the telephone network, assuring compatibility with all other 103 type modems. Because of the pureness of the sine wave, the STAR modem exceeds even the stringent harmonic requirements of all CCITT countries.

CARRIER DETECT

To assure accurate teleprocessing connections, the carrier detect circuitry prevents the modem from attempting to opcrate when excessive noise would produce errors or cause marginal operation. The circuitry also has a special amplitude sensor that prevents chatter when the received signal fades.

EXCLUSIVE ACOUSTIC CHAMBERS

The exclusive triple seal of Livermore's new flat mounted cups locks the handset into the acoustic chamber yielding superior acoustic isolation and mechanical cushioning. Designed to adapt to most common handsets used throughout the world, the STAR offers the utmost in flexibility and transmission reliability.

SELF TEST

The self test feature on the STAR allows the user to verify total operation of the acoustic modem by using the terminal in the full duplex mode. No need for remote assistance in diagnosing terminal or modem problems.

Utilizing the experience gained from building high quality couplers for over twelve years, Livermore has designed a coupler superior to any in its class for cost efficiency in industrial, commercial, business or home situations. You can see why we call it the STAR!

SPECIFICATIONS

Data Rate. 0 to 300 baud.
Compatibility. Bell 103 and 113; CCITT.
Transmit Frequencies.* Originate −1070 Hz/Space, 1270 Hz/Mark; Answer - 2025 Hz/Space, 2225 Hz/Mark
Receive Frequencies.* Originate - 2025 Hz/Space, 2225 Hz/Mark; Answer 1070 Hz/Space, 1270 Hz/Mark.
Frequency Stability. ± 0.3 percent.
Receiver Sensitivity. −50 dBm ON, −53 dBm OFF.
Transmit Level. −15 dBm.
Modulation. Frequency shift keyed (FSK).
Carrier Detect Delay. 1.2 seconds ON; 120 msec OFF.
EIA Terminal Interface. Compatible with RS 232 specifications.
Teletype Interface. 20 milliampere current loop.
Optional Interfaces. IEEE 488; TTL; TTY 43.

***International (CCITT) frequencies available.**

Switches. Originate/Off/Answer; Full Duplex/Test/Half Duplex.
Indicators. Transmit Data, Receive Data, Carrier Ready, Test.
Environmental. Ambient operating temperature 5° C. to 50° C. Relative humidity 10 to 90 percent (non-condensing).
Power. Suppled by 24 VAC/150 MA UL/CSA listed wall-mount transformer. Input 115 VAC, 2.5 watts. (A 220 VAC, 50 Hz adaptor is available upon request.)
Dimensions. 10"x 4"x 2"
Weight. 1.75 lbs. (2.2 lbs. shipping weight including AC adaptor.)
Warranty. Two years on parts and labor, excluding the AC adaptor which carries th manufacturer's warranty.

QT COMPUTER SYSTEMS INC.

IEEE S-100 COMPATIBLE

EXPANDABLE + DYNAMIC MEMORY (16K to 64K)

- + Works With Cromenco Systems
- + Uses 3242 Refresh Chip
- + 4 Layers Mean A Quiet Board
- + Bank Selectable Write Protect
- + Phantom Output Disable
- + Switch Selectable Output Disable

Bare Board	$ 49.95	32K Kit	$369.95	48K A&T	$494.95
16K Kit	$295.95	32K A&T	$419.95	64K Kit	$519.95
16K A&T	$345.95	48K Kit	$444.95	64K A&T	$569.95

Z + 80 CPU

- + 1K Ram On Board + 2 Programmable Timers
- + Switch Selectable 2 or 4 MHZ
- + Power On Jump to On-Board 1K or 2K EPROM (2708-2716-2732) Can be Addressed on any 1K, 2K or 4K Boundary
- + Programmable Baud Rate Selection (110 to 9600)
- + On-Board EPROM May be Used in Shadow Mode, Allowing Full 64K RAM to be Used
- + On-Board USART for Synchronous or Asynchronous RS-232 Operation (On-Board Baud Rate Generator)

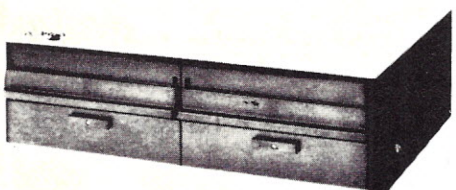
Bare Board	$ 45.00	A&T	$229.95
Kit	$169.95	1K Memory Kit	$ 12.00

CLOCK CALENDAR +

- + Time of Day in Hours, Minutes and Seconds
- + 24 Hour Time Format
- + Month and Day Date Function
- • Simple Read Instructions Allow Simple Interface to Basic, CPM, Etc.
- + Will Run With 4 MHZ Processors
- + Can be Located at any Group of 4 I/O Port Addressed

Bare Board $45.00 Kit $99.95 A&T $149.95

LOBO 8" DISK DRIVE CABINET



New from Lobo, a dual Cabinet with power supply, and internal data cable hook-up.

- Cabinet accepts 2 801R, 800R, FD120, or FD200 style disk drives.
- Power Supply for 2 drives.
- Assembled, tested and guaranteed by Lobo Drives.
- Shipping Weight 30 lbs.

LBO — DUAL 8 PCS..........$329.00

BUY CABINET AND DRIVES AND SAVE

WITH 1 DRIVE
LBO-801R-1PSC $775.00

WITH 2 DRIVES
LBO-801R-2PSC $1250.00

DISC DRIVE ONLY

SHU-801R.......$499.00

EXTERNAL DATA CABLES

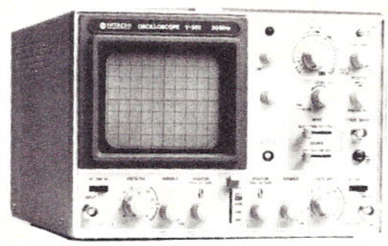
CARDEDGE TO CARDEDGE
PRI-50CE-CE $19.95

CARDEDGE TO SOCKET
PRI-50CE-SKT $19.95

MEMORY HEADQUARTERS

2716 16K 5 Volt only EPROM . . $22.00 ea.	10/$200.00
2708 8K 450ns EPROM . . . . 8/$55.00	$8.50 ea.
2114-3L 1Kx4 300ns Low Power . 8 $45.00	100 + $4.50 ea.
5257-3L 4Kx1 300ns Low Power . . 8/$55.00	100 + $5.25 ea.
2102AL-2 L/P 250ns in lots of 20 . 1.25 ea.	100 + 1.10 ea.

Hitachi V302



30MHZ DUAL TRACE OSCILLOSCOPE

LIST 945.00 - **SALE $798.00**

- TV sync-separater circuit
- High-sensitivity 1mV/div (5MHz)
- Sweep-time magnifier (10 times)
- Z-axis input (intensity modulation)
- Signal delay line
- X-Y operation
- Trace Rotation
- Complete with 2 probes
- CHI, CH2, DUAL, ADD, DIFF, Vertical Deflection Modes
- V152 Dual Trace 15MHZ - no delay sweep

LIST 695.00

SALE $595.00

✓49

PRIORITY ONE ELECTRONICS

16723K Roscoe Blvd. Sepulveda, CA 91343

ORDER TOLL FREE 1-800-423-5633 except CA., AK., HI., CALL (213) 894-8171

Terms: Visa, MC, BAC, Check, Money Order, U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00 Prepaid U.S. orders less than $75.00 include 5% shipping and handling. MINIMUM $2.50. Excess refunded. Just in case ... please include your phone no. Prices subject to change without notice. We will do our best to maintain prices thru Aug 1980.
*SOCKET and CONNECTOR prices based on GOLD, not exceeding $500 per oz.

*Sale Prices are for prepaid orders only credit card orders will be charged appropriate freight

FOR MORE INFORMATION SEE OUR 52 PAGE AD in JANUARY BYTE OR SEND $1.00 FOR CATALOG
• Sale Prices are for prepaid orders only • Quantities are limited, subject to prior sale • CREDIT CARD ORDERS WILL BE CHARGED APPROPRIATE FREIGHT

SEND $1.00 FOR 52 PAGE CATALOG

SEND $1.00 FOR 52 PAGE CATALOG

ORDER TOLL FREE (800) 423-5633

DIGITAL RESEARCH COMPUTERS

(214) 271-3538

32K S-100 EPROM CARD

NEW!

$74.95 KIT

USES 2716's

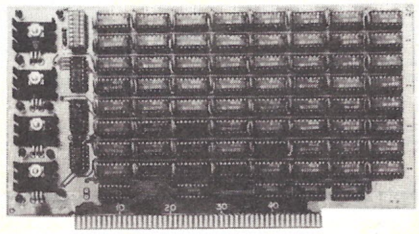
Blank PC Board - $34

ASSEMBLED & TESTED ADD $30

SPECIAL: 2716 EPROM's (450 NS) Are $19.95 EA. With Above Kit

KIT FEATURES:
1. Uses +5V only 2716 (2Kx8) EPROM's.
2. Allows up to 32K of software on line!
3. IEEE S-100 Compatible.
4. Addressable as two independent 16K blocks.
5. Cromemco extended or Northstar bank select.
6. On board wait state circuitry if needed.
7. Any or all EPROM locations can be disabled.
8. Double sided PC board, solder-masked, silk-screened.
9. Gold plated contact fingers.
10. Unselected EPROM's automatically powered down for low power.
11. Fully buffered and bypassed.
12. Easy and quick to assemble.

8K LOW POWER RAM KIT-S 100 BUSS



21L02 (450 NS RAMS!)

$119.50 KIT

ASSEMBLED & FULLY BURNED IN ADD $35

Thousands of computer systems rely on this rugged, work horse, RAM board. Designed for error-free, NO HASSLE, systems use

ADD $10 FOR 4 MHZ

Blank PC Board w/Documentation - **$29.95**
Low Profile Socket Set - **$13.50**
Support IC's (TTL & Regulators) - **$9.75**
Bypass CAP's (Disc & Tantalums) - **$4.50**

FOR 4MHZ SALE! **LOW POWER - 300NS 2114 RAM SALE!** **8 FOR $44**

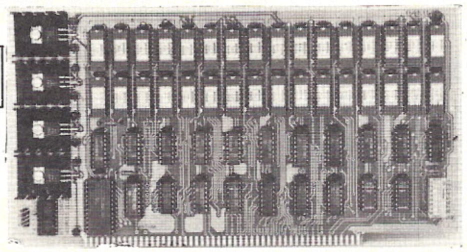
4K STATIC RAM'S. MAJOR BRAND, NEW PARTS.
These are the most sought after 2114's, LOW POWER and 300NS FAST.
8 FOR $44

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!

$225 KIT

FOR 4MHZ ADD $10



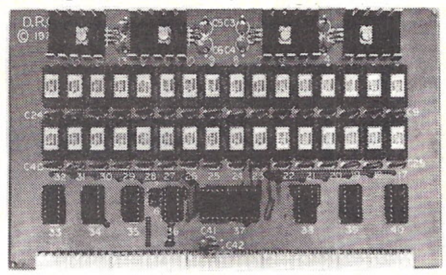
KIT FEATURES:
1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 1.5 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA-**$33**
LOW PROFILE SOCKET SET-**$12**
SUPPORT IC'S & CAPS-**$19.95**
ASSEMBLED & TESTED-**ADD $35**

OUR #1 SELLING RAM BOARD!

16K STATIC RAM SS-50 BUSS

PRICE CUT!



$229 KIT

FULLY STATIC!

FOR 2MHZ ADD $10

FOR SWTPC 6800 BUSS!

ASSEMBLED AND TESTED - $35

KIT FEATURES:
1. Addressable on 16K Boundaries
2. Uses 2114 Static Ram
3. Fully Bypassed
4. Double sided PC Board. Solder mask and silk screened layout.
5. All Parts and Sockets included
6. Low Power: Under 1.5 Amps Typical

BLANK PC BOARD—**$26** COMPLETE SOCKET SET—**$12**
SUPPORT IC'S AND CAPS—**$19.95**

NEW! STEREO! NEW! S-100 SOUND COMPUTER BOARD

At last, an S-100 Board that unleashes the full power of two unbelievable General Instruments AY3-8910 NMOS computer sound IC's. Allows you under total computer control to generate an infinite number of special sound effects for games or any other program. Sounds can be called in BASIC, ASSEMBLY LANGUAGE, etc.

COMPLETE KIT! $84.95 (WITH DATA MANUAL)

BLANK PC BOARD W/DATA $31

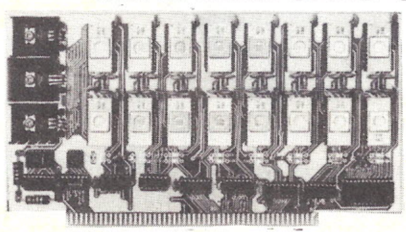
KIT FEATURES:
* TWO GI SOUND COMPUTER IC'S.
* FOUR PARALLEL I/O PORTS ON BOARD.
* USES ON BOARD AUDIO AMPS OR YOUR STEREO.
* ON BOARD PROTO TYPING AREA.
* ALL SOCKETS, PARTS AND HARDWARE ARE INCLUDED.
* PC BOARD IS SOLDERMASKED, SILK SCREENED, WITH GOLD CONTACTS.
* EASY, QUICK, AND FUN TO BUILD. WITH FULL INSTRUCTIONS.
* USES PROGRAMMED I/O FOR MAXIMUM SYSTEM FLEXIBILITY.

Both Basic and Assembly Language Programming examples are included.

SOFTWARE:

SCL™ is now available! Our Sound Command Language makes writing Sound Effects programs a SNAP! SCL™ also includes routines for Register-Examine-Modify, Memory-Examine-Modify, and Play-Memory. SCL™ is available on CP/M* compatible diskette of 2708 or 2716. Diskette - **$24.95** 2708 - **$19.95** 2716 - **$29.95** Diskette includes the source. EPROM'S are ORG at E000H.

16K EPROM CARD-S 100 BUSS



$59.95 KIT

BLANK PC BOARD - $28

USES 2708's!

Thousands of personal and business systems around the world use this board with complete satisfaction. Puts 16K of software on line at **ALL TIMES**! Kit features a top quality soldermasked and silk-screened PC board and first run parts and sockets. Any number of EPROM locations may be disabled to avoid any memory conflicts. Fully buffered and has WAIT STATE capabilities.

ASSEMBLED AND FULLY TESTED — ADD $30

OUR 450 NS 2708'S ARE $8.95 EA. WITH PURCHASE OF KIT

RCA CMOS COMPUTER CHIP SET

INCLUDES:

1-CDP1802CD CPU	1-CDP1861CD VIDEO IC
2-CDP1822CE 256 x 4 RAM	1-CDP1862CE COLOR GEN.
1-CDP1858CE 4 BIT LATCH	1-CDP1863CE SOUND GEN.

COMPLETE SET $45 LIMITED QTY

NEW! G.I. COMPUTER SOUND CHIP

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit Microprocessor. Contains: 3 Tone Channels, Noise Generator, 3 Channels of Amplitude Control. 16 bit Envelope Period Control, 2-8 Bit Parallel I/O. 3 D to A Converters, plus much more! All in one 40 Pin DIP. Super easy interface to the S-100 or other busses.

SPECIAL OFFER: $14.95 each Add $3 for 60 page Data Manual.

Digital Research Computers
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $1.00 postage, we pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. 90 Day Money Back Guarantee on all items. Orders over $50. add 85¢ for insurance.

***TRADEMARK OF DIGITAL RESEARCH.** **NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.**

✓ Reader Service index—page 241

kilobaud Peterborough NH 03458

MICROCOMPUTING T.M. LIST OF ADVERTISERS

***This advertiser prefers to be contacted directly.**

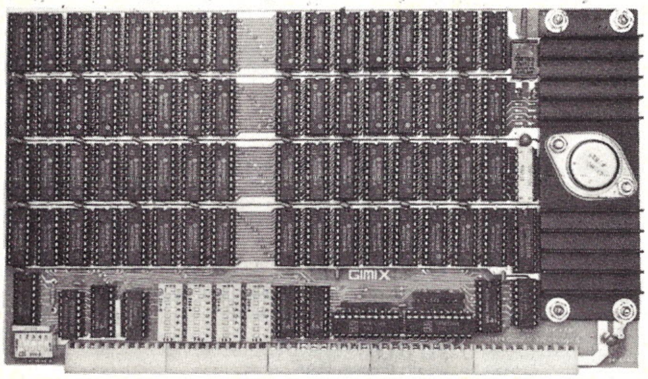
LOOK

WHAT'S COOKING on the FIFTY BUS

32K STATIC RAM BOARDS

Designed for use with:

★ Existing SS50 Systems ★ SS50C Extended Address Systems



- Assembled
- Burned In
- Tested

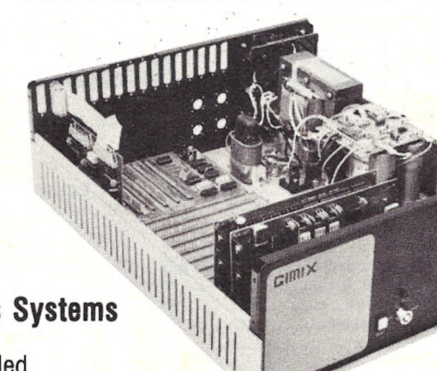
16K . . . $328.12
24K . . . $438.14
32K . . . $548.15

16K and 24K Versions are socketed for 32K and require only additional 2114's for expansion.

FEATURES:

- Decoding for 4 Extended Address Lines (allows memory decoding up to 1 megabyte)
- DIP-switch to set extended addressing or disable it
- 4 separate 8K blocks, addressable to any 8K boundary by DIP-switch
- Each 8K block may be individually disabled
- Write protect either of two 16K sections
- Low power consumption — uses 2114L low power RAMS
- Fully Socketed
- Gold Bus Connectors
- Guaranteed 2MHz operation

THE CLASSY CHASSIS



$898.19

- 25 amp (5V) ferro-resonant constant voltage power supply,
- Heavy weight aluminum cabinet with 3 position key switch, fan, and provisions for two 5'' disk drives;
- 6800/6809 Mother Board, fifteen 50 pin and eight DIP-switch addressable 30 pin slots (gold plated pins), fully decoded;
- Baud rate generator on I/O section of Mother Board.

I/O BOARDS

for the 30 PIN BUS:

1 Port Serial . . . . (RS 232 or 20MA, current loop)	**$ 88.41**
2 Port RS 232 Serial	**128.43**
2 Port Parallel	**88.42**

for the 50 PIN BUS:

8 Port RS 232 Serial	**288.40**
8 Port RS 232 Serial . . . . with on board Baud Rate generator.	**318.46**
8 Port Parallel	**198.45**

AND NOW . . . GIMIX OFFERS YOU A Choice of 6800 or 6809 CPU CARDS

You can order your system to fit your needs or select one of the below featured systems. Please contact the factory for further information and availability.

Add as much memory as you need using **GIMIX Static RAM Cards** for the utmost in reliability.

32K 6800 SYSTEM $1,694.59
Includes: Chassis, 6800 CPU, 32K RAM BOARD, I/O card

32K 6809 SYSTEM $1,844.69
Includes: Chassis, 6809 CPU, 32K RAM BOARD, I/O card

32K 6809 PLUS SYSTEM $1,994.79
Includes: Chassis, 32K RAM BOARD, I/O Card, and features our 6809 PLUS CPU Card with the Time of Day Clock option with battery back-up installed, as well as the 6840 Timer Package that provides 3 independent 16 bit counters.

This system also allows the following options to be added at additional cost:

- Battery back-up of the 1K RAM by substituting CMOS parts.
- A 9511 or 9512 Arithmetic Processor.
- GIMIX or SWTP Dynamic Address Translators.

BOTH 6809 SYSTEMS FEATURE OUR NEW TERMINAL BASED GMXBUG 09 SYSTEM MONITOR

GMXBUG 09 includes advanced debugging tools, utility, and memory manipulation routines.

Both 6809 Systems:

★ Can be reconfigured to allow use of other system monitors (OS-9 and SBUG-E)

★ Include 1K of Scratchpad RAM on the CPU

★ Allow optional software switching of system monitors.

2MHz 6809's at slight additional cost when they become available.

EXPORT NOTES:

For 50Hz 230V C.V. POWER SUPPLY . . . . Add $30.00
80 x 24 VIDEO BOARDS — Specify Format (No Added Charge)

On Orders under $250.00 for a Single Board, or Chips, please Add $30.00 Handling and we will ship Air Mail Prepaid. On all other orders we will ship via Emery Air Freight Collect, and we will charge no handling. All orders must be prepaid in U.S. Funds. Please note that foreign checks have been taking about eight weeks for collection, so we would advise wiring money or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, Account #73-32033. Visa or Master Charge also accepted.

FACTORY PRIME STATIC RAMS

2114L 450 ns . . $5.90 300 ns . . $6.40 200 ns . . $6.90

4044 450 ns . . $5.90 250 ns . . $6.90

ADD $5.00 HANDLING ON ORDERS UNDER $200.00

Phone, write, or see your dealer for details and prices on our broad range of Boards and Systems for the SS50/SS50C bus and our AC Power Control Products for all computers.

GIMIX INC.

✓22

The Company that delivers
Quality Electronic products since 1975.

1337 WEST 37th PLACE, CHICAGO, IL 60609
(312) 927-5510 • TWX 910-221-4055

Do more than ever before–spend less than you planned

Heath makes the All-In-One Computer more versatile

Many satisfied customers know Heath takes the risk out of buying a balanced computer system. With the Heathkit All-In-One Computer, you get 16K Random Access Memory (expandable to 48K), keyboard, video terminal and floppy disk system – together in one self-contained, compact unit – for up to hundreds of dollars less than comparable systems.

Heath now makes the All-In-One Computer more versatile than ever! The new Heathkit H77 Floppy Disk System gives the All-In-One even more data storage and recall capacity. Combined, the All- In-One and H77 Floppy Disk give you up to 300K bytes of on-line data storage – enough to hold entire files. You can mount operating system and program disks at the same time, to make computing even faster.

You can run programs written in MICROSOFT™ BASIC™ and Assembly Languages, and all current software written for the popular Heathkit H8 Computer.

Heath User's Group (HUG) will share with you a library of over 500 programs to make your computer serve you in ways you never imagined.

There's no better way to learn about computer systems – and save money – than by building one yourself.

Concise, easy-to-follow Heathkit assembly manuals show you the way, from start to finish. And a nationwide network of service centers protects your computer investment.

Join the Heathkit computer family today – and pocket the savings!

For complete details on Heathkit computer systems, as well as nearly 400 other electronic kits for your home, work or pleasure, send today for your free, value-packed Heathkit catalog. Or pick up your copy at the nearest Heathkit Electronic Center.

Heathkit®

✓236

SEND FOR FREE CATALOG | **VISIT YOUR HEATHKIT STORE**

Write to: HEATH COMPANY,
DEPARTMENT 351-684
BENTON HARBOR, MI 49022

In the U.S. and Canada, visit your nearby Heathkit Electronic Center where Heathkit products are also displayed, sold and serviced. See the white pages of your phone book. In the U.S., Heathkit Electronic Centers are units of Veritechnology Electronics Corporation.

CP-184

The Businessman's Business System

MSI Business Computer Systems offer flexibility and expandability unmatched by any other microcomputer system, large or small. Our SDOS operating system is totally device independent and supports up to four users. This means that you can start with a single user, dual drive, floppy disk system today, and add up to 80 megabytes of hard disk with additional workstations tomorrow. As your business grows, your MSI system grows with you—and your software won't become obsolete.

Perform text processing tasks at one workstation while entering sales orders on another. Add a third workstation in inventory control and a fourth in accounting. That's expandability!!!

- MSI Inventory Software, with complete Bills of Material, provides a complete inventory control and management system for manufacturers.
- Complete manufacturing forecasting, with production pick lists, allows automatic adjustment of component inventory levels.
- All transactions resulting in any change to the inventory data base are written to audit trail files listing date, time, operator's name, inventory item, and the changes which were made.
- Sales Order Entry/Accounts Receivable Software displays customer balances and credit standing as new orders are entered. Correct product prices and descriptions are obtained from inventory files if desired.
- Invoices are generated automatically as orders are shipped. Customer statements, with aged accounts receivable, are printed on demand.
- Purchase Order Entry/Accounts Payable Software optionally link to inventory program, in order to easily visualize inventory items which are on order.
- General Ledger programs link to the accounts receivable and accounts payable modules for easy updates and posting.
- If your business is expanding and you would like to know how an MSI Computer System can help you make it more profitable, call or write Midwest Scientific Instruments, 220 W. Cedar, Olathe, Kansas 66061, (913) 764-3273, TWX 910 749 6403 (MSI OLAT), TELEX 42525 (MSI A OLAT).

Small Computers For Big Jobs

MSI Midwest Scientific Instruments